Machine Translation Summit XVI

http://www.mtsummit2017.org

Proceedings of MT Summit XVI
Vol.1 Research Track

Sadao Kurohashi, Pascale Fung, Editors

MT Summit XVI
September 18 – 22, 2017 -- Nagoya, Aichi, Japan

Proceedings of MT Summit XVI,

Vol. 1: MT Research Track

Sadao Kurohashi (Kyoto University)
&
Pascale Fung (Hong Kong University of Science and Technology), Eds.

Co-hosted by

International Association for Machine Translation
http://www.eamt.org/iamt.php

Asia-Pacific Association for Machine Translation
http://www.aamt.info

Graduate School of Informatics, Nagoya University
http://www.is.nagoya-u.ac.jp/index_en.html

Research Program Co-chairs

Sadao Kurohashi	Kyoto University
Pascale Fung	Hong Kong University of Science and Technology

Research Program Committee

Yuki Arase	Osaka University
Laurent Besacier	LIG
Houda Bouamor	Carnegie Mellon University
Hailong Cao	Harbin Institute of Technology
Michael Carl	Copenhagen Business School
Daniel Cer	Google
Boxing Chen	NRC
Colin Cherry	NRC
Chenhui Chu	Osaka University
Marta R. Costa-jussa	Universitat Politècnica de Catalunya
Steve DeNeefe	SDL Language Weaver
Markus Dreyer	Amazon
Kevin Duh	Johns Hopkins University
Andreas Eisele	DGT, European Commission
Christian Federmann	Microsoft Research
Minwei Feng	IBM Watson Group
Mikel L. Forcada	Universitat d'Alacant
George Foster	National Research Council
Isao Goto	NHK
Spence Green	Lilt Inc
Eva Hasler	SDL
Yifan He	Bosch Research and Technology Center
Philipp Koehn	Johns Hopkins University
Roland Kuhn	National Research Council of Canada
Shankar Kumar	Google
Anoop Kunchukuttan	IIT Bombay
Jong-Hyeok Lee	Pohang University of Science & Technology
Gregor Leusch	eBay

William Lewis	Microsoft Research
Mu Li	Microsoft Research
Lemao Liu	Tencent AI Lab
Qun Liu	Dublin City University
Klaus Macherey	Google
Wolfgang Macherey	Google
Saab Mansour	Apple
Daniel Marcu	Amazon
Jonathan May	USC Information Sciences Institute
Arul Menezes	Microsoft Research
Haitao Mi	Alipay US
Graham Neubig	Carnegie Mellon University
Matt Post	Johns Hopkins University
Fatiha Sadat	UQAM
Michel Simard	NRC
Katsuhito Sudoh	Nara Institute of Science and Technology
Christoph Tillmann	IBM Research
Masao Utiyama	NICT
Taro Watanabe	Google
Andy Way	ADAPT, Dublin City University
Deyi Xiong	Soochow University
Francois Yvon	LIMSI/CNRS
Rabih Zbib	Raytheon BBN Technologies
Jiajun Zhang	Institute of Automation Chinese Academy of Sciences
Bing Zhao	SRI International

Contents

Empirical Study of Dropout Scheme for Neural Machine Translation

Xiaolin Wang xiaolin.wang@nict.go.jp
Masao Utiyama mutiyama@nict.go.jp
Eiichiro Sumita eiichiro.sumita@nict.go.jp
Advanced Translation Research and Development Promotion Center,
National Institute of Information and Communications Technology, Japan

Abstract

Dropout has lately been recognized as an effective method to relieve over-fitting when training deep neural networks. However, there has been little work studying the optimal dropout scheme for neural machine translation (NMT). NMT models usually contain attention mechanisms and multiple recurrent layers, thus applying dropout becomes a non-trivial task. This paper approached this problem empirically through experiments where dropout were applied to different parts of connections using different dropout rates. The work in this paper not only leads to an improvement over an established baseline, but also provides useful heuristics about using dropout effectively to train NMT models. These heuristics include which part of connections in NMT models have higher priority for dropout than the others, and how to correctly enhance the effect of dropout for difficult translation tasks.

1 Introduction

Neural machine translation (NMT), as a new technology emerged from the field of deep learning, has improved the quality of automated machine translation into a significantly higher level compared to statistical machine translation (SMT) (Wu et al., 2016; Sennrich et al., 2017; Klein et al., 2017). State-of-the-art NMT models, like many other deep neural networks, typically contain multiple non-linear hidden layers. This makes them very expressive models, which is critical to successfully learning very complicated relationships between inputs and outputs (Devlin et al., 2014). However, as these large models have millions of parameters, they tend to over-fit during training phrases.

Dropout has recently been recognized as a very effective method to relieve over-fitting. Dropout was first proposed for feed-forward neural networks by Hinton et al. (2012). Dropout was then successfully applied to recurrent neural networks by Pham et al. (2014) and Zaremba et al. (2014). Dropout outperforms many traditional approaches to over-fitting, including early stop of training, introducing weight penalties of various kinds such as L1 and L2 regularization, decaying learning rate and so on.

Reported state-of-the-art NMT systems all adopt dropout during training phrases (Wu et al., 2016; Klein et al., 2017), but their paper have not provided many details about how dropout was applied. The optimal way to apply dropout to NMT models is non-trivial. Strictly speaking, NMT is an application of deep neural networks, so it can directly benefit from the advance of deep neural networks. However, two facts make NMT models stand out from normal deep neural networks. First, NMT models contain recurrent hidden layers in order to gain

the ability of operating on sequences. As contrast, deep neural networks in face recognition and phoneme recognition are feed-forward as they take separate inputs (Povey et al., 2011; Krizhevsky et al., 2012). Second, NMT models contain a novel attention mechanism to manage a memory of an input sequence, as this sequence can become quite long (Bahdanau et al., 2014). This attention mechanism involves calculations such as *weighted mean*, which is rarely used in normal deep neural networks.

As far as we know, there has been no focused study on how to apply dropout effectively to NMT models. This motivated the work presented in this paper, which aimed at finding out the optimal dropout scheme for training NMT models. Because the architecture of NMT models is complicated, we took an empirical approach. We trained many NMT models by verifying the subsets of connections that dropout were applied to, and using different dropout rates for different connections. We tried to find out the optimal dropout scheme through answering the following two questions,

- what part of NMT models should be applied dropout to during training phrases;
- how to correctly set the dropout rates for training NMT models.

The following of this paper is structured as: the section 2 reviews related works on dropout, then the section 3 describes the method that we adopted to search for optimal dropout scheme, after that the section 4 presents the results of the experiments, and in the end the section 5 concludes this paper with a description on our future work.

2 Related Works

Hinton et al. (2012) and Srivastava et al. (2014) proposed the method of dropout to relieve overfitting when training feed-forward neural networks. They worked with a variety of feed-forward neural networks each of which is established for a certain task. Different dropout schemes were applied to these neutral networks in order to obtain optimal results.

- MNIST is a standard toy data set of handwritten digits(LeCun et al., 2010). The best network had two layers of 8 195 rectified linear units. Dropout was applied to the input units with a rate $p = 0.2$, and the hidden units with $p = 0.5$.
- CIFAR-10 and CIFAR-100 are tiny natural images (Krizhevsky and Hinton, 2009); Street View House Numbers is a data set of images of house numbers collected by Google Street View (Netzer et al., 2011). The best network had three convolutional layers followed by two fully connected hidden layers. The output of the last fully-connected layer was fed to a softmax which produced a distribution over the class labels. All hidden units were rectified linear units. Each convolution layer was followed by a max-pooling layer. Dropout was applied to all layers including the input layer with $p = \{ 0.10, 0.25, 0.25, 0.50, 0.50, 0.50 \}$.
- ImageNet is a large collection of natural images (Deng et al., 2009). The best network had five convolutional layers followed by three fully connected hidden layers. The numbers of units in each layers were 253 440, 186 624, 64 896, 64 896, 43 264, 4 096, 4 096, and 1 000. The output of the last fully-connected layer was also fed to a softmax layer which produced a distribution over all class labels (Krizhevsky et al., 2012). Dropout was only applied to the first two fully connected hidden layers with $p = 0.50$. The reason might be that the network was very big . The author claimed that dropout roughly doubled the number of iterations required to converge. It can be inferred that applying dropout to all layers might not be feasible because of the time cost on training.

- TIMIT is a standard speech benchmark for clean speech recognition (Garofolo et al., 1993). The best network had six layers. Dropout was applied to the input layer with $p = 0.2$ and the hidden layers with $p = 0.5$.
- Alternative Splicing is a data set of RNA features for predicting alternative gene splicing (Xiong et al., 2011). The best network had two layers of 1024 hidden units. Dropout was applied to the input layer with $p = 0.2$ and the hidden layers with $p = 0.5$.

Their experiments were limited on feed-forward networks, which were much simpler than NMT networks. However, their experimental results suggest two useful heuristics about achieving good performance from dropout. First, dropout need to be applied to all layers of networks. Second, the optimal dropout rates for different type of layers such as input layers and hidden layers are different.

Zaremba et al. (2014) studied how to correctly apply dropout to recurrent neural networks such as long short-term memory (LSTM) units. They proposed that dropout should only be applied vertically, that is, be applied to non-recurrent connections. They argued that applying dropout to recurrent layers will amplify noise, as discussed by Bayer et al. (2013). They performed experiments on a variety of tasks, one of which was a machine translation task. Their NMT model contained no attention mechanism, which was like a recurrent language model trained on concatenations of source sentences and their translations (Sutskever et al., 2014). It had four layers of 1 000 LSTM units, three embedding layers (source language input embedding, target language input and output embedding) and a softmax layer. Dropout was applied to the connections between input-embedding-to-LSTM, LSTM-to-LSTM, LSTM-to-output-embedding [1] with $p = 0.2$. Their experiments were performed on a selected subset of the WMT 2014 English to French data set containing 340M French words and 304M English words Schwenk et al. (2011). Experimental results showed that dropout improved BLEU scores from 25.90 to 29.03, while still lost to a BLEU score of 33.30 achieved by a phrase-based SMT system named LIUM.

Wu et al. (2016) achieved a great improvement of translation quality through NMT when compared to their previous phrase-based production systems. They adopted a deep LSTM network that had eight encoder layers, eight decoder layers and a attention layer. They claimed that they adopted a dropout scheme similar to the method in Zaremba et al. (2014), but no further details were provided, especially on how dropout was applied to the attention layer.

Klein et al. (2017) released an NMT toolkit named OpenNMT. It implemented a network architecture similar with the one proposed by Luong et al. (2015). OpenNMT outperformed the SMT system of Moses (Koehn et al., 2007) and a few other NMT systems including GroundHog and Blocks in our pilot experiments. Therefore we took it as an important baseline in this paper. OpenNMT did not follow the dropout scheme of Zaremba et al. (2014) to apply dropout to all non-recurrent layers. Instead, OpenNMT applied dropout only to non-top encoding and decoding LSTM layers, and output hidden states(see section 3.2 for details). This dropout scheme seems arbitrary, but it did performed quit well in experiments. Solving this puzzle is one of the main motivations of this paper.

3 Methods

This section first presents the architecture of the NMT model that we adopted, which is one of state-of-the-art attention-based encoder-decoder NMT model. After that, this section analyzes that architecture and provides a list of connections in the architecture that are appropriate for applying dropout to. In the end, this section describes the method that we adopted to search for the optimal dropout scheme for training NMT models.

[1] According to the source code in `https://github.com/wojzaremba/lstm`.

3.1 Neural Machine Translation

The essence of a machine translation system is modeling the conditional probability of a translation given a source sentence. In encoder-decoder NMT models, it can be formalized using chain's rule as,

$$log\, p(\mathbf{y}|\mathbf{x}) = \sum_{j=1}^{m} log(p(y_j|y_1^{j-1}, \mathbf{s}) \tag{1}$$

where $\mathbf{x}$=$(x_1, \ldots, x_n, \langle \mathrm{EOS} \rangle)$ is a source sentence and $\langle \mathrm{EOS} \rangle$ is a end-of-sentence token; $\mathbf{y}$=$(y_1, \ldots, y_m)$ is a translation; $\mathbf{s}$ is an representation of $\mathbf{x}$ produced by the encoder. Note that $\mathbf{x}$ sometimes reverse its order to help the decoder to translate from the beginning of a source sentence.

In this paper, we adopted a compact stacking recurrent architecture as the encoder-decoder (illustrated by figure 1), which was proposed by Luong et al. (2015). This architecture assumes that equation 1 is factorized into a calculable form as,

$$\begin{aligned} log\, p(\mathbf{y}|\mathbf{x}) &= \sum_{j=1}^{m} log(p(y_j|H_o^{\langle j \rangle}) \\ &= \sum_{j=1}^{m} \log(\mathrm{softmax}_{y_j}(\tanh(W_o H_o^{\langle j \rangle} + B_o))) \qquad (2) \\ H_o^{\langle j \rangle} &= \mathcal{F}_{\mathrm{att}}(H_s, H_t^{\langle j \rangle}), \qquad (3) \end{aligned}$$

where H_s is a source-side hidden state produced by the top recurrent layer of the encoder; H_t is a target-side hidden state produced by the top recurrent layer of the decoder; H_o is a output hidden state produced by the attention model $\mathcal{F}_{\mathrm{att}}$; the superscript $^{\langle j \rangle}$ is a target-side timestamp; W_o and B_o are the matrix and bias of output embedding; and $\mathrm{softmax}_{y_j}$ means selecting the dimension from the output of the softmax which corresponds to y_j in one-hot encoding.

We adopted a global attention model (Luong et al., 2015) as $\mathcal{F}_{\mathrm{att}}$ (illustrated by figure 2). This attention model first calculates an alignment weight as,

$$\begin{aligned} a_{st}^{\langle ij \rangle} &= \mathrm{softmax}(\mathcal{F}_{\mathrm{a}}(H_s^{\langle i \rangle}, H_t^{\langle j \rangle})) \\ &= \frac{e^{\mathcal{F}_{\mathrm{a}}(H_s^{\langle i \rangle}, H_t^{\langle j \rangle})}}{\sum_{i=1}^{n} e^{\mathcal{F}_{\mathrm{a}}(H_s^{\langle i \rangle}, H_t^{\langle j \rangle})}}, \qquad (4) \\ \mathcal{F}_{\mathrm{a}}(H_s^{\langle i \rangle}, H_t^{\langle j \rangle}) &= H_s^{\langle i \rangle \top} W_{\mathrm{a}} H_t^{\langle j \rangle}, \qquad (5) \end{aligned}$$

where $\mathcal{F}_{\mathrm{a}}$ is a scoring function for alignment, which is composed of a linear mapping and a dot product; and W_{a} is a matrix for linearly mapping target-side hidden states into a space which is comparable to the source-side.

Then the attention model calculates translation contexts as,

$$\begin{aligned} C_s^{\langle j \rangle} &= \sum_{i=1}^{n} a_{st}^{\langle ij \rangle} H_s^{\langle i \rangle} \qquad (6) \\ C_{st}^{\langle j \rangle} &= [C_s; H_t^{\langle j \rangle}], \qquad (7) \end{aligned}$$

where $C_s^{\langle j \rangle}$ is a source-side context, and $C_{st}^{\langle j \rangle}$ is a context derived from both source and target sides through concatenating.

In the end, the attention model calculates an output hidden state as,

$$H_o^{\langle j \rangle} = W_c C_s t^{\langle j \rangle}, \tag{8}$$

where W_c is a matrix for linearly mapping a context into an output hidden state.

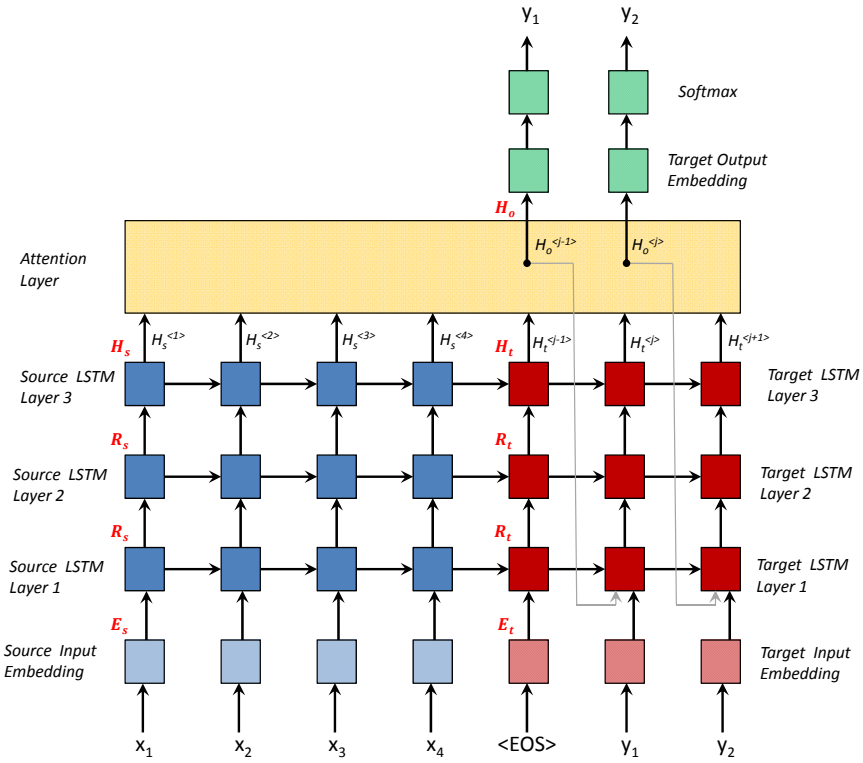


Figure 1: Network Architecture of Neural Machine Translation

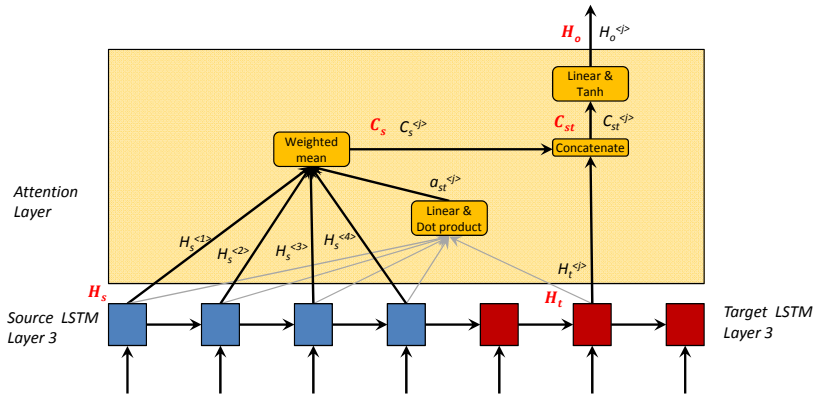


Figure 2: Illustration of Attention Model

3.2 Appropriate Connections for Dropout

Dropout should be applied vertically according to the previous study by Zaremba et al. (2014). This means that dropout should only be applied to non-recurrent connections. Therefore, there are nine connections in the NMT model that are appropriate for applying dropout to. The figures 1 and 2 annotate the corresponding variables with red colored symbols. For the sake of convenience, this paper views each connection in the NMT model as a variable. Applying dropout to a variable means applying dropout to those non-recurrent connections that transport the variable.

A detailed list of the nine variables appropriate for dropout in the NMT model is as follows,

- E_s input embedding of source-side;
- R_s hidden states of source-side non-top recurrent layers;
- H_s hidden states of source-side top recurrent layer;
- E_t input embedding of target-side;
- R_t hidden states of target-side non-top recurrent layers;
- H_t hidden states of target-side top recurrent layers;
- H_o hidden states of output;
- C_s translation context of source-side;
- C_{st} concatenate of source and target-side translation contexts.

The impact of applying dropout to these nine variables are not independent. Especially, the four variables involved in the attention model, including H_s, C_s, H_t and C_{st}, are closely related to each other. This is because the operators of *weighted mean* (the equation 6) and *concatenate* (the equation 7) reserve the effect of dropout. In other words, the dropout that is applied to their input will propagate onto their output.

Because of the complicated relations among these variables, the optimal way to apply dropout becomes a non-trivial task. For example, there are two choices if we want to make NMT model robust to source representations. One is to apply dropout to H_s, which will affect the calculation of alignment weight a_{st}, weighted mean C_s, and concatenated context C_st. The other one is to apply dropout to C_s, which will leave the calculation of alignment weight a_st untouched. It is quite difficult to predict the end-to-end performances of these two choices.

3.3 Search for Optimal Dropout Scheme

We decomposed the task of searching for optimal dropout scheme into two steps. The first step was to search for an optimal combination of variables for applying dropout to. The second step was to search for optimal dropout rates for each variable in the optimal combination.

Note that training NMT models was very time consuming, so exploring the full search space was impossible. Therefore we sometimes terminated searches early if one result was particularly good. We were aware that pruning search space might cause results not to be globally optimal. However, it made searching for optimal dropout scheme a feasible task.

3.3.1 Search for Optimal Combination of Variables

As described above, there are nine variables in the NMT model which are appropriate for dropout. The established toolkit of OpenNMT chooses to apply dropout to three of the nine variables, including R_s, R_t and H_o (Klein et al., 2017). This decision seems quite arbitrary.

Therefore, it was meaningful to find out which combination of these nine variables lead to good performance.

We took a heuristic greedy search to find the optimal combination of variables. We gradually increased the size of combinations. We started by applying dropout to only one variable. Then we tried applying dropout to two variables. After that we continued by applying dropout to three variables, and continued like this. In each stage we aimed to find the best combination of a given size. We generally tried adding variables to the promising combinations in the previous stage.

3.3.2 Search for Optimal Dropout Rates

Hinton et al. (2012) and Srivastava et al. (2014) showed that optimal dropout rates for different layers of feed-forward networks are different. Because the nine variables appropriate for dropout in the NMT model play different roles, they may have different optimal dropout rates. Therefore, we explored applying different dropout rates to the variables in the optimal combination, which was found in the first step.

We took a grid search to find optimal dropout rates . In each step, we tried increasing or decreasing the dropout rate of one variable by a fixed amount such as 0.05. We then chose the update on a variable which maximized the performance.

4 Experiments

This section first describes our experimental settings, then presents the results of searching for optimal combination of variables for applying dropout to, after that presents the results of searching for optimal dropout rates, and in the end compares our optimal dropout schemes with baselines.

4.1 Experimental Settings

Two corpora were used in our experiments (see the table 1). The first corpus was from the shared task of NIST Open Machine Translation 2006 Evaluation (OpenMT Chinese-to-English) [2]. We first removed the UN and the traditional Chinese data sets from the NIST-2006 constraint training resources. Then we performed word segmentation on the Chinese text using the Stanford word segmenter (Tseng et al., 2005), and performed tokenization on the English text using the scripts provided in (Koehn, 2005). The data sets of NIST Eval 2004 and 2005 were used as a development set. The data set of NIST Eval 2016 was used as a test set.

The second corpus was the Basic Travel Expression Corpus (BTEC) (Takezawa et al., 2002). We used the in-house English-to-Japanese corpus which contains about 463k sentences. We randomly selected 2 000 sentences as a development set, and selected another 2 000 sentences as a test set. The sentences left over were used as a training set. The English text was also tokenized using the scripts provided in (Koehn, 2005), and the Japanese text was segmented into words using the toolkit of Mecab (Kudo, 2005).

The wordpiece model was adopted to deal with rare words for NMT (Wu et al., 2016). This approach breaks all words, especially the rare ones, into subword units that are like to occur more often in a training corpus. Therefore, these rare words become translatable by NMT models. Byte Pair Encoding was adopted to train segmentation models (Gage, 1994). This method allows for the representation of an open vocabulary through a fixed-size vocabulary of variable-length character sequences, making it very suitable for close-vocabulary systems like NMT (Sennrich et al., 2015). In this paper, we adopted a vocabulary size of 16k according to our pilot experiments.

[2] `http://www.itl.nist.gov/iad/mig//tests/mt/2006/`

Data Set	# Sentences	# Words		Vocabulary	
		Source	Target	Source	Target
Corpus: OpenMT Chinese-to-English					
Training	442,967	12,265,072	13,444,927	178,832	130,249
Development	2,679	72,869	87,369† (346,231‡)	NA	NA
Test	1,664	37,827	46,207† (193,214‡)	NA	NA
Corpus: BTEC English-to-Japanese					
Training	458,894	3,664,481	4,193,101	27.757	36,308
Development	2,000	16,148	18,451	NA	NA
Test	1,664	15,866	18,048	NA	NA

Table 1: Experimental Corpora. † the first reference; ‡ totally four references.

The phrase-based SMT toolkit of Moses was adopted as a baseline (Koehn et al., 2007). The Moses' models were trained in a conventional settings. The toolkit of SRILM was adopted to train 5-gram language models on target languages (Stolcke, 2002). The toolkit of Giza++ (Och, 2003) was adopted to perform word alignment. Then the training scripts provided by Moses were employed to build translation models. Then the systems were tuned with MERT on development sets (Och, 2003).

The NMT toolkit of OpenNMT was adopted as another baseline. OpenNMT outperformed Moses and a few other NMT systems including GroundHog and and Blocks in our pilot experiments. Therefore we took it as a baseline.

Different dropout schemes were tested using our C++ implementation of NMT, named CytonMT. The implementation utilizes NVIDIA's native libraries including CUDA, CUBLAS and CUDNN to gain efficiency on NVIDIA's GPUs. CytonMT adopts the network architecture proposed by Luong et al. (2015), which is similar with the one implemented by OpenNMT.

NMT models were trained with a similar setting as Luong et al. (2015). The stacking LSTM models had four layers of 1 024 cells, and 1 024-dimensional embedding. The parameters of neural networks were initialized in [-0.1, 0.1]. The parameters were trained with stochastic gradient descending algorithm. The gradient normalized gradient was re-scaled when it exceeded 5.

A simple adaptive learning rate schedule was employed to ensure that models with heavy dropout were fully trained. The training started with a learning rate of 1. If the perplexity on the development set did not decrease after an epoch, the learning rate started to decay by 0.5 per epoch. After that if the perplexity did not decrease in two continuous epochs, the training phrase was terminated. The maximum number of epochs was unlimited, while training usually finished around 20 epochs.

Translation performances of difference methods were measured by BLEU. BLEU was calculated on the lower-cased English words in the task of OpenMT Chinese-to-English, and was calculated on the Japanese characters in the task of BTEC English-to-Japanese.

4.2 Results of Searching for Optimal Combination of Variables

A group of experiments were performed following the method described in the section 3.3.1. The table 2 presents the results of the experiments. Main experiments were performed on the OpenMT corpus, and the BTEC corpus were used for confirming the findings.

The experiments were categorized into seven stages (separated by horizontal lines in the table), as the number of variables in the combination were gradually increased. In each stage, we aimed to find the best combination of given size (annotated by bold fonts in the table). We

generally tried adding variables to the promising combination in the previous stage. Because training NMT models s time consuming, we terminated the stage early if the best one was clear.

Two observations can be made from these experimental results. First, two special variables C_t and C_{st} are not suitable for dropout. They are called special because they are the output of *weighted mean* and *concatenate*, so they inherit the effect of dropout from the input. Experiments 21, 22, 29 and 30 show that the applying dropout to C_t or C_{st} leads to poorer performance than applying dropout to upstream variables H_s or H_t.

Second, among the seven remaining variables, the priority of applying dropout to each variable can be formulated as a chain,

$$R_t \succ R_s \succ H_o \succ E_s \succ H_s \succeq H_t \succeq E_t \tag{9}$$

where $\succ$ means context-ed superior, and $\succeq$ means context-ed superior or equal, with respects to dropout. $x_1 \succ \ldots \succ x_k \succ x_{k+1}$, means that given the context that $(x_1, \ldots, x_{k-1})$ have already been applied dropout to, applying dropout to x_k is superior to applying dropout to x_{k+1}. In other words, applying dropout to $(x_1, \ldots, x_{k-1}, x_k)$ outperforms applying dropout to $(x1 \ldots x_{k-1}, x_{k+1})$.

The priority chain of the equation 9 confirms that the OpenNMT's dropout scheme is an effective one, because R_t, R_s and H_o are the three top variables. Besides, the chain also suggests two other effective dropout schemes. The optimal-1 is to apply dropout to R_t, R_s, H_o and E_s. The experiment 17 shows that the cross entropy on the development set decreases by adding E_s into the OpenNMT's dropout scheme. The optimal-2 is to apply dropout to all the seven remaining variables. The experiments 23 – 27 show that adding any one or two from H_t, H_s and E_t into the optimal-1 brings little improvement, but adding all three variable into optimal-1 reduces the cross entropy.

4.3 Results of Searching for Optimal Dropout Rates

In this subsection, we aimed to refine the optimal dropout schemes found in the last subsection by using different dropout rates for each variable. We applied the grid search method described in the section 3.3.2. The table 3 and 4 presents the results of refining the optimal-2 on the OpenMT corpus and refining the optimal-1 on the BTEC corpus, respectively.

Unexpectedly, the experimental results on both corpora show that no changes on the dropout rates can improve the performances. Therefore, p = 0.3 is an optimal dropout rate for training the NMT model that we adopt.

4.4 Comparison with Baselines

In this section, we compared the optimal dropout schemes found in our study with baselines. Three baselines were employed. The first was Moses – one of the state-of-the-art phrase-based SMT systems. The second was the NMT toolkit of OpenNMT. The third was our implementation of CytonMT using the OpenNMT's dropout scheme. Among these three baseline, the third was the most accurate. The table 5 presents the results of the experiments.

Three observations can be made from the experimental results. First, on the OpenMT corpus, both the optimal-1 and the optimal-2 outperform the baselines (the experiments 1–5). In addition, the optimal-2 outperforms the optimal-1. This validates the effectiveness of optimal-1 and the optimal-2.

Second, on the BTEC corpus, the performances of the three dropout schemes are close. The different behaviors on the two corpora may be caused by the fact that the OpenMT corpus is more difficult than the BTEC corpus with respects to translation, as its sentences are longer and its vocabulary is larger. From the baseline to the optimal-1 and optimal-2, the strength of dropout gradually increases since more and more variables are being applied dropout to.

No.	Method: Apply Dropout to	Cross Entropy: Training	Cross Entropy: Development
	Corpus: OpenMT Chinese-to-English		
1	E_s	1.52	2.65
2	E_t	2.16	3.46
3	R_s	2.20	3.61
4	$\boldsymbol{R_t}$	1.59	**2.58**
5	H_s	1.66	2.85
6	H_t	1.63	2.62
7	H_o	2.33	3.51
8	C_s	2.24	3.56
9	C_{st}	1.74	2.70
10	R_tE_s	1.57	2.53
11	$\boldsymbol{R_tR_s}$	1.53	**2.46**
12	R_tH_t	1.72	2.53
13	R_tH_o	1.53	2.55
14	$R_tR_sE_s$	1.49	2.50
15	$E_tR_sH_t$	1.74	2.53
16	$\boldsymbol{R_tR_sH_o}$ $^\natural$	1.50	**2.41**
17	$\boldsymbol{R_tR_sH_oE_s}$ †	1.59	**2.36**
18	$R_tR_sH_oE_t$	1.70	2.40
19	$R_tR_sH_oH_s$	1.70	2.37
20	$R_tR_sH_oH_t$	1.68	2.40
21	$R_tR_sH_oC_s$	1.74	2.45
22	$R_tR_sH_oC_{st}$	1.70	2.43
23	$R_tR_sH_oE_sE_t$	1.59	2.37
24	$R_tR_sH_oE_sH_s$	1.65	2.36
25	$\boldsymbol{R_tR_sH_oE_sH_t}$	1.70	**2.36**
26	$R_tR_sH_oE_sH_tE_t$	1.72	2.37
27	$\boldsymbol{R_tR_sH_oE_sH_tH_s}$	1.79	**2.36**
28	$\boldsymbol{R_tR_sH_oE_sH_sH_sE_t}$ ‡	1.79	**2.33**
29	$R_tR_sH_oE_sE_tH_tC_s$	1.845	2.39
30	$R_tR_sH_oE_sE_tC_{st}$	1.834	2.41
	Corpus: BTEC English-to-Japanese		
31	$R_tR_sH_oE_s$ †	0.81	1.12
32	$\boldsymbol{R_tR_sH_oE_sH_tH_sE_t}$ ‡	0.83	**1.11**

Table 2: Results of Applying Dropout to Different Combinations of Variables. The Dropout rate is $p = 0.3$. $^\natural$ dropout scheme of the toolkit OpenMT; † the optimal-1; ‡ the optimal-2.

No.	Dropout Rate							Cross Entropy	
	R_t	R_s	H_o	E_s	H_t	H_S	E_t	Training	Development
Corpus: OpenMT Chinese-to-English									
Ref.	0.30	0.30	0.30	0.30	0.30	0.30	0.30	1.79	**2.33**
1	**0.35**	0.30	0.30	0.30	0.30	0.30	0.30	1.79	2.38
2	**0.25**	0.30	0.30	0.30	0.30	0.30	0.30	1.96	2.43
3	0.30	**0.35**	0.30	0.30	0.30	0.30	0.30	1.85	2.36
4	0.30	**0.25**	0.30	0.30	0.30	0.30	0.30	1.77	2.38
5	0.30	0.30	**0.35**	0.30	0.30	0.30	0.30	1.73	2.35
6	0.30	0.30	**0.25**	0.30	0.30	0.30	0.30	1.77	2.35
7	0.30	0.30	0.30	**0.35**	0.30	0.30	0.30	1.80	2.37
8	0.30	0.30	0.30	**0.25**	0.30	0.30	0.30	1.74	2.36
9	0.30	0.30	0.30	0.30	**0.35**	0.30	0.30	1.81	2.34
10	0.30	0.30	0.30	0.30	**0.25**	0.30	0.30	1.79	2.34
11	0.30	0.30	0.30	0.30	0.30	**0.35**	0.30	1.81	2.37
12	0.30	0.30	0.30	0.30	0.30	**0.25**	0.30	1.78	2.34
13	0.30	0.30	0.30	0.30	0.30	0.30	**0.35**	1.89	2.41
14	0.30	0.30	0.30	0.30	0.30	0.30	**0.25**	1.81	2.37

Table 3: Results of Using Different Dropout Rates on the optimal-2.

No.	Dropout Rate				Cross Entropy	
	R_t	R_s	H_o	E_s	Training	Development
Corpus: BTEC English-to-Japanese						
Ref.	0.30	0.30	0.30	0.30	0.81	**1.12**
9	**0.35**	0.30	0.30	0.30	0.73	1.13
10	**0.25**	0.30	0.30	0.30	0.73	1.13
11	0.30	**0.35**	0.30	0.30	0.82	1.12
12	0.30	**0.25**	0.30	0.30	0.77	1.12
13	0.30	0.30	**0.35**	0.30	0.73	1.13
14	0.30	0.30	**0.25**	0.30	0.76	1.12
15	0.30	0.30	0.30	**0.35**	0.80	1.13
16	0.30	0.30	0.30	**0.25**	0.79	1.12

Table 4: Results of Using Different Dropout Rates on the optimal-1.

No.	System	Dropout Scheme	Cross Entropy		BLEU	
			Train.	Dev.	Dev.	Test
Corpora: OpenMT Chinese-to-English						
1	Moses	NA	NA	NA	32.12	31.11
2	OpenNMT	baseline$^\natural$	1.62	2.42	39.96	39.11
3	CytonMT	baseline$^\natural$	1.50	2.41	40.07	39.21
4	CytonMT	optimal-1†	1.59	2.36	40.39	39.38
5	CytonMT	optimal-2‡	1.79	**2.33**	40.35	**39.89**
Corpora: BTEC English-to-Japanese						
6	Moses	NA	NA	NA	52.09	50.77
7	OpenNMT	baseline$^\natural$	0.63	1.15	52.35	52.38
8	CytonMT	baseline$^\natural$	0.81	1.12	52.49	**52.46**
9	CytonMT	optimal-1†	0.81	1.12	52.58	52.44
10	CytonMT	optimal-2‡	0.83	**1.11**	52.63	52.33

Table 5: Comparison with Baseline Methods. $^\natural$ baseline: OpenNMT's method, applying dropout to R_t, R_s and H_o. † optimal-1: applying dropout to R_t, R_s, H_o and E_s. ‡ optimal-2: applying dropout to all variables of R_t, R_s, H_o, E_s, H_t, E_t and H_s but exclude C_s and C_{st}. The drop rate is fixed as p = 0.3 .

Therefore, for easy translation tasks, the baseline or the optimal-1 is sufficient; while for difficult tasks, the optimal-2 is recommended.

Third, all the NMT systems trained with dropout clearly outperformed the SMT system. This indicates that the baseline dropout scheme is effective. This also confirms the description in the section 1 of this paper.

5 Conclusion

In this paper, we performed an empirical study on dropout scheme for training NMT models. We started the study by analyzing the architecture of NMT models, and found out the appropriate variables for applying dropout to. We then run two groups of experiments to find out the optimal combination of these variables and the optimal dropout rate.

Two main questions raised in the introduction can be answered through our study. The first question is what part of NMT models should be applied dropout. The priority of the variables in NMT models is

$$R_t \succ R_s \succ H_o \succ E_s \succ H_s \succeq H_t \succeq E_t.$$

This chain suggests some effective dropout schemes, including the OpenNMT's scheme, the optimal-1, and the optimal-2 (section 4.2). Note that heavy dropout scheme will increase the required number of epochs in training phase. If a translation task is difficult, and training time is sufficient, the optimal-2 is recommended (section 4.4). We empirically find that training NMT models using OpenNMT's dropout scheme usually converges within the 13 epochs [3] which is quite efficient, while using the optimal-2 usually requires around 20 epochs.

The second question is how to correctly set dropout rate for training NMT models. It is found that the dropout rate p = 0.3 is optimal for the NMT model that we adopt (section 4.3).

In the future, we plan to explore two topics related to dropout for NMT, including max-norm regularization (Srivastava et al., 2014) and drop connections (Wan et al., 2013).

[3] the default setting of the toolkit OpenNMT

References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.

Bayer, J., Osendorfer, C., Korhammer, D., Chen, N., Urban, S., and van der Smagt, P. (2013). On fast dropout and its applicability to recurrent networks. *arXiv preprint arXiv:1311.0701*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. M., and Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *ACL (1)*, pages 1370–1380. Citeseer.

Gage, P. (1994). A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., and Pallett, D. S. (1993). DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon technical report n*, 93.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. M. (2017). OpenNMT: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*, volume 5, pages 79–86.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kudo, T. (2005). Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab. sourceforge. net/*.

LeCun, Y., Cortes, C., and Burges, C. J. (2010). MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Pham, V., Bluche, T., Kermorvant, C., and Louradour, J. (2014). Dropout improves recurrent neural networks for handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 285–290. IEEE.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi speech recognition toolkit. In *IEEE Workshop on Automatic Speech Recognition and Understanding*.

Schwenk, H., Lambert, P., Barrault, L., Servan, C., Afli, H., Abdul-Rauf, S., and Shah, K. (2011). LIUM's SMT machine translation systems for WMT 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 464–469. Association for Computational Linguistics.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., L"aubli, S., Miceli Barone, A. V., Mokry, J., and Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Takezawa, T., Sumita, E., Sugaya, F., Yamamoto, H., and Yamamoto, S. (2002). Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *LREC*, pages 147–152.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., and Manning, C. (2005). A conditional random field word segmenter. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*, volume 171. Jeju Island, Korea.

Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1058–1066.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xiong, H. Y., Barash, Y., and Frey, B. J. (2011). Bayesian prediction of tissue-regulated splicing using RNA sequence and cellular context. *Bioinformatics*, 27(18):2554–2562.

Zaremba, W., Sutskever, I., and Vinyals, O. (2014). Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

A Target Attention Model for Neural Machine Translation

Hideya Mino † ‡ mino.h-gq@nhk.or.jp
Andrew Finch † andyfinch@hotmail.com
Eiichiro Sumita † eiichiro.sumita@nict.go.jp
† National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, JAPAN
‡ Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Meguro-ku, 13, 152-8550, JAPAN

Abstract

Neural Machine Translation (NMT) with an attention mechanism has shown promising results by utilizing word alignments between the source and target sentences. Typically, training of NMT proceeds token-by-token on the target side, where each token is predicted using only a vector representing the current hidden-state, and the previous token. However, this strategy has serious shortcomings originating the lack of information about the partial target sequence hypothesis; specifically, this can lead to source tokens being translated multiple times or remaining untranslated. To alleviate this problem, we introduce a target-side attention mechanism to exploit the generated target sequence of tokens more effectively. We calculate a target-side context vector using a recurrent neural network and feed it to an attention mechanism so that the decoder can pay more or less attention to each token in the partially generated target sequence when predicting the next target token. Experiments on three different English-to-Japanese translation tasks show improvements of 0.6-1.5 BLEU points.

1 Introduction

Recently, Neural Machine Translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014) has been growing in popularity due to its capacity to model the translation process end-to-end within a single probabilistic model, and its potential for higher performance compared to existing phrase-based statistical machine translation (SMT) (Koehn, 2004). There are some unique features of NMT models which pose significant challenges for machine translation. One is that NMT systems exploit Long Short-Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997) (or the similar Gated Recurrent Units (GRUs) (Cho et al., 2014)) which allow the systems to capture long-distance dependencies better than vanilla RNNs. Another is the attention mechanism, whereby the decoder can attend directly to localized information from the source sequence of tokens for generating the target sequence (Bahdanau et al., 2015; Luong et al., 2015). NMT systems are generally trained to maximize the likelihood of generating the target sequence of tokens given the source sequence. In practice, each target token is generated conditioned on the vector representing the current hidden-state of the model, and the previously generated target token.

NMT, however, has a serious drawback in that some input tokens are unnecessarily translated or mistakenly left untranslated (Tu et al., 2016). Our hypothesis is that this is mainly

because the hidden state of the LSTM decoder is not sufficiently representing all the information concerning the generated target sequence of tokens. Our work therefore endeavors to alleviate this drawback by explicitly handing a summary of the target sequence generated so far, at each step in the decoding process. Although an LSTM is able to provide the function of a long-term memory, the prediction of target tokens in a state-of-the-art NMT model (Bahdanau et al., 2015) heavily depends on two factors: the source-side context vectors with focus provided by an attention model, and a target language model implicitly learned by the LSTM decoder. This NMT model fails to exploit the generated target-side information, which is useful to avoid over- and under-translation problems. If target words translated in the past is accumulated appropriately to the LSTM decoder, they are less likely to be translated again, and new target word which is not translated yet should be generated. Because of ignoring the information of the sequence of previously generated target tokens, unnecessarily translated words and mistakenly untranslated words are generated. To alleviate the lack of target-side information in the LSTM decoder, we propose to add a target-side context vector directly into the NMT model. The target-side context vector is generated with the attention mechanism, which selects the relevant target tokens for predicting the next target token. We show empirically that the addition of this target-side context vector significantly improves the performance of an NMT system on three different English-to-Japanese translation tasks.

2 Related Work

There is much recent work on augmenting attention-based NMT systems with additional features. One focus is the use of the monolingual data (Sennrich et al., 2016; Gülçehre et al., 2015). Gülçehre et al. (2015) incorporated a large language model into an attention-based NMT system to allow the effective use of target-side monolingual data. Another focus is in designing better decoding strategies (Luong et al., 2015; Tu et al., 2016; Mi et al., 2016; Liu et al., 2016; Mi et al., 2016; Tu et al., 2017). Tu et al. (2017) proposed to augment a direct model's decoding objective with a reverse translation model. Liu et al. (2016) proposed translating in both a left-to-right and a right-to-left direction and seeking a consensus. Tu et al. (2016) introduced a coverage vector to keep track of the attention history, which encourages the attention-based NMT system not to translate source words for multiple times (i.e., avoiding over-translation) and to translate more untranslated source words (i.e., avoiding under-translation). Mi et al. (2016) also dealt with the coverage problem.

We also tackle on the over- and under-translation problems. Our approach differs from those of Tu et al. (2016) and Mi et al. (2016) in that they utilize only source-side attention history, whereas our approach also exploits the sequence of target tokens generated.

3 Neural Machine Translation with a Source Attention Model

Our method is based on NMT with attention (Bahdanau et al., 2015), which generates the target sentence $\boldsymbol{y} = (y_1, ..., y_M)$ from the source sentence $\boldsymbol{x} = (x_1, ..., x_N)$ of length N, as illustrated in Figure 1 (note: we use bold script to denote sequences hereafter). The attention-based model consists of two components, an encoder and a decoder. The encoder reads the source sentence $\boldsymbol{x}$ and encodes it into hidden states $\boldsymbol{h} = (h_1, ..., h_N)$. The hidden states are produced using a bidirectional RNN, which concatenates a forward and a backward sequences, as

$$h_j = \begin{bmatrix} \overrightarrow{h}_j \\ \overleftarrow{h}_j \end{bmatrix} \tag{1}$$

where

$$\overrightarrow{h}_j = e_1(x_j, \overrightarrow{h}_{j-1}), \overleftarrow{h}_j = e_2(x_j, \overleftarrow{h}_{j+1}). \tag{2}$$

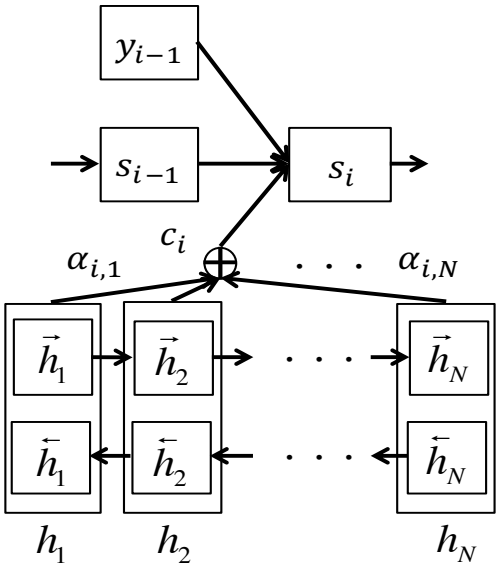


Figure 1: Encoder-decoder NMT architecture with source attention

e_1 and e_2 are nonlinear functions. Bahdanau et al. (2015) used a GRU (Cho et al., 2014) for e_1 and e_2. Each hidden state, represented as a single vector, includes not only the lexical information at its source position, but also information about the unbounded length of the left and right context. Then, the decoder predicts the target sentence $\boldsymbol{y}$ using a conditional probability calculated as

$$p(y_i|\boldsymbol{y}_{1,i-1}, \boldsymbol{x}) = f_1(y_{i-1}, s_i, c_i) \tag{3}$$

where $\boldsymbol{y}_{1,i-1}$ is a partial translation $(y_1, ..., y_{i-1})$, f_1 is implemented as a feedforward neural network with a softmax output layer, s_i is a hidden state of the RNN, and c_i is a context vector derived from the source sentence. The hidden state s_i of the target RNN is computed by

$$s_i = g_1(s_{i-1}, y_{i-1}, c_i) \tag{4}$$

where g_1 is a nonlinear function analogous to e_1 or e_2. The context vector c_i is computed as a convex sum of the hidden states h_j of Equation (1):

$$c_i = \sum_{j=1}^{N} \alpha_{i,j} h_j \tag{5}$$

where $\alpha_{i,j}$ is a scalar weight of each hidden state h_j computed by

$$\alpha_{i,j} = \frac{\exp\{a(s_{i-1}, h_j)\}}{\sum_{k=1}^{N} \exp\{a(s_{i-1}, h_k)\}} \tag{6}$$

where a is a feedforward neural network with a single hidden layer. The attention mechanism is driven by this $\alpha_{i,j}$, which shows how well the input context at the j-th word and the output word at the i-th position match. The objective is to jointly maximize the conditional probability

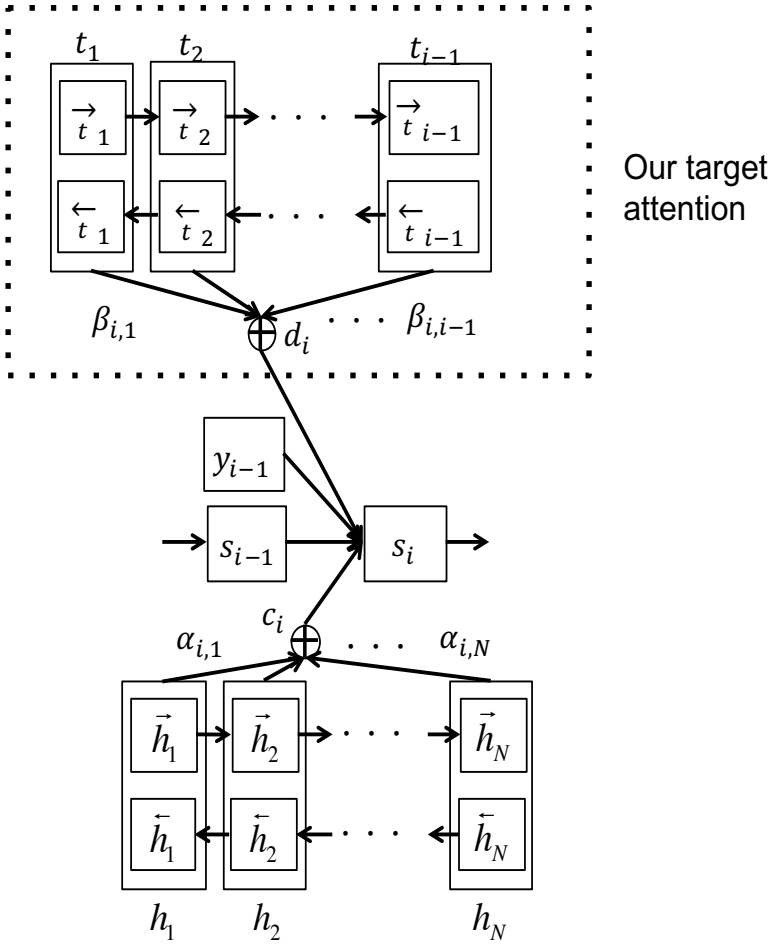


Figure 2: The proposed encoder-decoder NMT architecture with both source and target attention

for each generated target word as

$$\theta^* = \arg\max_{\theta} \sum_{k=1}^{K} \sum_{i=1}^{M_k} \log p(y_i^k | \mathbf{y}_{1,i-1}^k, \mathbf{x}^k, \theta) \tag{7}$$

where $(\mathbf{x}^k, \mathbf{y}^k)$ is the k-th training pair of sentences, and M_k is the length of the k-th target sentence $\mathbf{y}^k$.

4 Adding a Target Attention Model

Attention-based NMT usually uses an LSTM for decoding from an encoded source sentence as a whole, and a single previous target token as in Equation (4). Intuitively, the encoded source sentence and the generated sequence of target tokens are both indispensable for predicting the next target token. Although LSTMs have been shown to be capable of predicting the next token in a sequence given a compressed representation of the preceding sequence, this process becomes considerably more difficult when compressing long sequences (Liu et al., 2016). To

strengthen the information provided by the generated target sequence of tokens, our model adds a target-side context vector to the input of the LSTM decoder at each decoding step, as shown in Figure 2. In this model, a representation of the generated target sequence is explicitly made available to the decoder at each step instead of implicitly relying on the LSTM to maintain it.

In addition, the semantics each token of the generated target sequence depends on its context. The LSTM model produces a vector that contains compressed information representing an unfocused summary of the whole generated target sequence. In order to allow the model to focus on salient contexts, we use a mechanism for focusing on the relevant parts of the already-generated target sequence for generating the current target token, along with a bidirectional layer to provide the model with the a good representation of the target.

The proposed method is implemented as a target-side attention model constructed analogously to the source-side attention model, where the attention ranges over the partially generated target token sequence. More formally, the partial translation $\boldsymbol{y}_{1,i-1}$ is encoded into a sequence of hidden states $\boldsymbol{t}_{1,i-1}$, which are produced using a bidirectional RNN, as

$$t_k = \begin{bmatrix} \overrightarrow{t}_k \\ \overleftarrow{t}_k \end{bmatrix} \quad (1 \leq k \leq i-1) \tag{8}$$

where

$$\overrightarrow{t}_i = e_3(y_i, \overrightarrow{t}_{i-1}), \overleftarrow{t}_i = e_4(y_i, \overleftarrow{t}_{i+1}). \tag{9}$$

e_3 and e_4 are nonlinear functions as in Equation (2). Then, the decoder predicts the target sentence with a conditional probability as

$$p(y_i|\boldsymbol{y}_{1,i-1}, \boldsymbol{x}) = f_2(y_{i-1}, s_i, c_i, d_i) \tag{10}$$

where f_2 is a probability estimator as in Equation (3) and newly introduced d_i is a predicted target-side context vector. The computation of the hidden state s_i is also modified as

$$s_i = g_2(s_{i-1}, y_{i-1}, c_i, d_i) \tag{11}$$

where g_2 is a nonlinear function as in Equation (4). The context vector d_i is computed as a convex sum of the hidden states $\boldsymbol{t}_{1,i-1}$:

$$d_i = \sum_{k=1}^{i-1} \beta_{i,k} t_k \tag{12}$$

where $\beta_{i,k}$ is also a scalar weight of each hidden state t_k as below:

$$\beta_{i,k} = \frac{\exp\{b(s_{i-1}, t_k)\}}{\sum_{k=1}^{i-1} \exp\{b(s_{i-1}, t_k)\}} \tag{13}$$

where b is a feedforward neural network analogous to a in Equation (6). $\beta_{i,k}$ gives a normalized score for each previous target token, which measures how the k-th target word is relevant to the prediction of the i-th target token. The objective is again to jointly maximize the likelihood as in Equation (7). Typically, the previous target token y_{i-1} used by the LSTM decoder is the true previous token when training, and a predicted previous token during decoding. In our experiments, we follow this practice, although there is evidence that using predictions during training would be beneficial (Bengio et al., 2015). Since our approach is orthogonal to that of Bengio et al. (2015), it would be possible to use both techniques in tandem.

Corpus	Training					Development			Test		
	Sents.	Word types		Avg. length		Sents.	Word types		Sents.	Word types	
		en	ja	en	ja		en	ja		en	ja
IWSLT'07	40k	9.4k	10k	9.3	12.7	0.5k	1.2k	1.3k	0.5k	0.8k	0.9k
NTCIR-10	717k	105k	79k	23.3	27.7	2.0k	5.0k	4.4k	0.5k	2.4k	2.1k
ASPEC	843k	288k	143k	22.1	23.9	1.8k	7.1k	6.3k	1.8k	7.0k	6.4k

Table 1: Data sets

5 Experiments

We evaluated the proposed method on three different English-to-Japanese translation tasks. As a baseline, we trained the attention-based NMT and the coverage-vector method (Tu et al., 2016). To confirm the effectiveness of the target-side bidirectional RNN in the proposed method, we also trained the proposed method with one direction RNN, from left to right.

5.1 Data and model parameters

The corpora we used were IWSLT'07 (Fordyce, 2007), NTCIR-10 (Goto et al., 2013), and ASPEC (Nakazawa et al., 2016). IWSLT'07 consists of spoken travel conversations, NTCIR-10 consists of patents, and ASPEC is in the domain of scientific publications. We constrained training sentences to have a maximum length of 40 to speed up the training.[1] As shown in Table 1, the data size of IWSLT'07 is smaller than the other corpora, and ASPEC has a greater lexical variety compared to the others. Each test sentence had a single reference translation. The English data was tokenized using the tokenization script included in the Moses decoder.[2] The Japanese data was tokenized with KyTea (Neubig et al., 2011).

5.2 Settings

The input and output of our model are sequences of one-hot vectors with dimensionality corresponding to the sizes of the source and target vocabularies. For NTCIR-10 and ASPEC, we replaced words of frequency less than 3 with the [*UNK*] symbol and excluded them from the vocabularies. As a result, the number of word types in NTCIR-10 turned out 60k for English and 50k for Japanese, and ASPEC contained 124k types for English and 79k for Japanese. Due to the limited memory of GPU, each source and target word was projected into a 200-dimensional continuous Euclidean space to reduce the dimensionality, the depth of the stacking LSTMs was 1 and hidden layer size was set to 300. Each model was optimized using Adam (Kingma and Ba, 2014) with the following parameters: $\alpha = 1e-3$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-8$. To prevent overfitting we used dropout (Srivastava et al., 2014) with a drop rate of $r = 0.5$ to the last layer of each stacking LSTM. All weight metrics of each model were initialized by sampling from a normal distribution of zero mean and 0.05 standard deviation. The gradient at each update is calculated using a minibatch of at most 100 sentence pairs and we ran for a maximum of 30 iterations for the entire training data. Training was early-stopped to maximize the performance on the development set measured by BLEU. We used a single Tesla K80 GPU with 12 GB of memory for the training. For decoding, we used beam search with a beam size of 10. The beam search was terminated when an end-of-sentence [*EOS*] symbol was generated.

The evaluation metric is case-insensitive BLEU (Papineni et al., 2002) calculated by the

[1] The proposed method takes approximately five times the training time, and three times the decoding time, relative to the baseline attention-based NMT. The proposed method with one direction RNN, instead of bidirectional RNN, takes approximately three times the training time, and three times the decoding time.

[2] http://statmt.org/moses/

System	IWSLT'07	NTCIR-10	ASPEC
source-attn	47.4	31.0	26.2
coverage-vector	47.7	31.4	25.8
source-and-target-attn (left-to-right)	48.0	31.5	26.4
source-and-target-attn (bidirectional)	48.3	32.3 †‡	27.7 †‡

Table 2: BLEU scores for the attention-based NMT (*source-attn*), the coverage vector method (Tu et al., 2016) (*coverage-vector*) and the proposed method (*source-and-target-attn*) with target-side bidirectional RNN (*bidirectional*) and target-side one directional RNN from left to right (*left-to-right*) (†: significantly better than *source-attn* ($p < 0.05$); ‡: significantly better than *coverage-vector* ($p < 0.05$).

System	IWSLT'07	NTCIR-10	ASPEC
source-attn	39 / 0.91	412 / 0.94	1178 / 0.91
coverage-vector	91 / 0.91	347 / 0.92	884 / 0.89
source-and-target-attn (left-to-right)	58 / 0.91	286 / 0.93	870 / 0.90
source-and-target-attn (bidirectional)	38 / 0.90	335 / 0.94	659 / 0.91

Table 3: Numbers of overtranslated words (left-side) and averages of the brevity penalty per sentence (right-side)

`multi-bleu.perl` script in the Moses toolkit. Statistical significance testing of the BLEU differences was performed using paired bootstrap resampling (Koehn, 2004) with 10,000 iterations. We also assessed the decrease in the over- and under-translation with two kinds of criteria. For the over-translation, we used a number of overtranslated words, which are unnecessarily translated though these are already translated in outputs. We simply counted the number of repeated phrases (length longer or equal than 2 words) for each sentence as in Mi et al. (2016). For the under-translation, we used an average of brevity penalty per sentence. The brevity penalty, which is part of BLEU, is to penalize predicted sentence that are shorter than the reference.

5.3 Results

Table 2 summarizes the results for all the three tasks. For the IWSLT'07 task, our model achieved 0.9, 0.6, and 0.3 BLEU point improvements compared with *source-attn*, *coverage-vector*, and *source-and-target-attn (left-to-right)*, respectively. For the NTCIR-10 task, our model achieved gains of 1.3, 0.9, and 0.8 BLEU points. For the ASPEC task, our model achieved gains of 1.5, 1.9, and 1.3 BLEU points. These results show that our proposed method is more effective than other baseline methods. The results for IWSLT'07 show less improvement than those for NTCIR-10 and ASPEC. The reason for this may be the length of the target. As shown in Table 1, the average length of sentence of IWSLT'07 is much shorter than NTCIR-10 and ASPEC. These results show that the proposed method seems to be more effective for the tasks with long sentences. The explanation is most likely analogous to the motivation for using a source-side attention model: an LSTM model without attention struggles to propagate necessary information over longer distances. Our target-side attention model explicitly facilitates this.

Table 3 shows the numbers of overtranslated words and the averages of the brevity penalty. The brevity penalty is 1.0 when the output length is longer than the reference translation's length. For IWSLT'07, there were no improvements. As mentioned earlier, we believe the cause is related to the fact that the sentences in this corpus are short; our method is most ef-

fective for longer sequences. For the other two tasks, our model seemed to be able to reduce the number of overtranslated words, also maintaining the target sequence length closer to that of the references. For NTCIR-10, though *source-and-target-attn (left-to-right)* greatly reduces the number of overtranslated words, the BLEU score is almost same as *coverage-vector*. It shows that *source-and-target-attn (left-to-right)* increases the number of mistranslated words and *source-and-target-attn (bidirectional)* is effective to decrease not only the number of overtranslated words but also the number of mistranslated words. Examples of outputs generated by each model are shown in Appendix A.

These analyses validate our contribution to the original motivation for this work, i.e., the proposed model is capable of effectively decreasing the number of mistakenly untranslated words and unnecessarily translations of the same word.

6 Conclusion

We introduced a focused summary of the target sequence generated so far into the decoding process in order to alleviate the problems of the over- and under-translation problems. Our empirical evaluation shows that the proposed method is effective in achieving substantial improvements in terms of translation quality consistently across three different tasks.

Acknowledgements

We are deeply grateful to Atsushi Fujita and anonymous reviewers for their suggestions and insightful comments on the early version of this paper.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR 2015)*.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. M. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NIPS 2015)*.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Fordyce, C. S. (2007). Overview of the 4th international workshop on spoken language translation iwslt 2007 evaluation campaign. In *Proceedings of the 4th International Workshop on Spoken Language Translation (IWSLT 2007)*.

Goto, I., Chow, K., Lu, B., Sumita, E., and Tsou, B. K. (2013). Overview of the patent machine translation task at the NTCIR-10 workshop. In *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR-10)*.

Gülçehre, Ç., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8).

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*.

Liu, L., Utiyama, M., Finch, A., and Sumita, E. (2016). Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.

Mi, H., Sankaran, B., Wang, Z., and Ittycheriah, A. (2016). Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*.

Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S., and Isahara, H. (2016). Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*.

Neubig, G., Nakata, Y., and Mori, S. (2011). Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2011)*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15.

Sutskever, I., Vinyals, O., and Le, Q. V. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS 2014)*.

Tu, Z., Liu, Y., Shang, L., Liu, X., and Li, H. (2017). Neural machine translation with reconstruction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016). Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.

Appendix A. Examples of outputs

We show examples of Japanese translation generated with each of the four models in Tables 2 and 3 with a source sentence and a reference. The words shown in bold letters are examples of over- or under-translation problems.

Examples of NTCIR-10

[Example 1]

Source sentence:

This fluctuation in the power supply voltage and reference voltage causes power source noise .

Reference:

このようにして**電源 電圧 (the power supply voltage)** や **基準 電圧 (the reference voltage)** が変動して電源ノイズを生じさせる。

Output with *source-attn*:

電源 電圧 (the power supply voltage) と **電源 電圧 (the power supply voltage)** との変動により、電源ノイズが発生する。

Output with *coverage-vector*:

電源 電圧 (the power supply voltage) の変動により、**電源 電圧 (the power supply voltage)** が変動し、**電源 電圧 (the power supply voltage)** が発生する。

Output with *source-and-target-attn (left-to-right)*:

電源 電圧 (the power supply voltage) および **基準 電圧 (the reference voltage)** の変動は、電源ノイズを発生する。

Output with *source-and-target-attn (bidirectional*):

電源 電圧 (the power supply voltage) と **基準 電圧 (the reference voltage)** との変動は、電源ノイズを発生する。

[Example 2]

Source sentence:

As shown in FIG . 5 , the drain current is also affected by the stress .

Reference:

図5に示したように**ドレイン 電流 (the drain current)** も応力の影響を受ける。

Output with *source-attn*:

５． ５に示すように、**ドレイン 電流 (the drain current)** の影響を受けることにより、**ドレイン 電流 (the drain current)** が影響を受ける。

Output with *coverage-vector*:

５ｖに示すように、**ドレイン 電流 (the drain current)** により**ドレイン 電流 (the drain current)** も影響を受ける。

Output with *source-and-target-attn (left-to-right)*:

図５に示すように、**ドレイン 電流 (the drain current)** は、応力によって**影響を与える (is affecting)**。

Output with *source-and-target-attn (bidirectional*):

図５に示すように、**ドレイン 電流 (the drain current)** は、応力によって**影響を受ける (is affected by)**。

Examples of ASPEC

[Example 1]

Source sentence:

> compatible solutes include polyols such as glycine betaine (betaine) , zwitterionic proline , pinitol , sorbitol , and mannitol .

Reference:

> 適合 溶質 と して は， グリシンベタイン （ ベタイン ）， プロ リン など の **両性 イオン (zwitterionic)**， **ピニトール (pinitol)**， **ソルビトール (sorbitol)**， マンニトール など の ポリオール 類 など が 知ら れて いる 。

Output with *source-attn*:

> 適合 溶質 は グリシンベタイン （ グリシンベタイン ）， **ピニトール (pinitol)**， **ピニトール (pinitol)**， **ピニトール (pinitol)**， **ピニトール (pinitol)**， **ピニトール (pinitol)** で ある 。

Output with *coverage-vector*:

> 適合 溶質 と して は， グリシンベタイン （ ベタイン ）， **ピニトール (pinitol)**， **ピニトール (pinitol)**， **ピニトール (pinitol)**， **ピニトール (pinitol)**， **ピニトール (pinitol)** で ある 。

Output with *source-and-target-attn (left-to-right)*:

> 適合 溶質 と して は グリシンベタイン （ ベタイン ）， **両性 イオン (zwitterionic)**， **ピニトール (pinitol)**， **ピニトール (pinitol)**， マンニトール など の ポリオール 類 が 含まれる 。

Output with *source-and-target-attn (bidirectional*):

> 適合 溶質 と して は， グリシンベタイン （ ベタイン ）， **ピニトール (pinitol)**， **ソルビトール (solbitol)**， マンニトール など の ポリオール を 含む 。

[Example 2]

Source sentence:

> the liquid-crystal film in the title was prepared on a glass substrate by uniaxial orientation and the amount of the transmitted light through a polarizing plate was measured .

Reference:

> 標題 液晶 薄膜 を **ガラス 基板 上 に (on a glass substrate)** 一 軸 配向 さ せて 形成 し， 偏光 板 を 経た 透過 光 量 を 測定 した 。

Output with *source-attn*:

> 標記 薄膜 を **ガラス 基板 上 に (on a glass substrate)** **ガラス 基板 上 に (on a glass substrate)** 調 製 し， 偏光 板 を 通過 する 透過 光 の 量 を 測定 した 。

Output with *coverage-vector*:

> 標記 薄膜 を **ガラス 基板 上 に (on a glass substrate)** **ガラス 基板 上 に (on a glass substrate)** 作製 し， 偏光 板 を 通過 する 透過 光 の 量 を 測定 した 。

Output with *source-and-target-attn (left-to-right)*:

> 標記 液晶 膜 を 一 軸 配向 に より **ガラス 基板 上 に (on a glass substrate)** 作製 し， 偏光 板 を 介した 透過 光 の 量 を 測定 した 。

Output with *source-and-target-attn (bidirectional*):

> 標記 液晶 を **ガラス 基板 上 に (on a glass substrate)** 一 軸 配向 に より 作製 し， 偏光 顕微 鏡 に より 透過 光 の 量 を 測定 した 。

Neural Pre-Translation for Hybrid Machine Translation

Jinhua Du jinhua.du@adaptcentre.ie
Andy Way andy.way@adaptcentre.ie
ADAPT, School of Computing, Dublin City University, Ireland

Abstract

Hybrid machine translation (HMT) takes advantage of different types of machine translation (MT) systems to improve translation performance. Neural machine translation (NMT) can produce more fluent translations while phrase-based statistical machine translation (PB-SMT) can produce adequate results primarily due to the contribution of the translation model. In this paper, we propose a cascaded hybrid framework to combine NMT and PB-SMT to improve translation quality. Specifically, we first use the trained NMT system to pre-translate the training data, and then employ the pre-translated training data to build an SMT system and tune parameters using the pre-translated development set. Finally, the SMT system is utilised as a post-processing step to re-decode the pre-translated test set and produce the final result. Experiments conducted on Japanese→English and Chinese→English show that the proposed cascaded hybrid framework can significantly improve performance by 2.38 BLEU points and 4.22 BLEU points, respectively, compared to the baseline NMT system.

1 Introduction

In recent years, NMT has made impressive progress (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015). The state-of-the-art NMT model employs an encoder–decoder architecture with an attention mechanism, in which the encoder summarizes the source sentence into a vector representation, the decoder produces the target string word by word from vector representations, and the attention mechanism learns the soft alignment of a target word against source words (Bahdanau et al., 2015). NMT systems have outperformed the state-of-the-art SMT model on various language pairs in terms of translation quality (Luong et al., 2015a; Bentivogli et al., 2016; Junczys-Dowmunt et al., 2016; Wu et al., 2016; Toral and Sánchez-Cartagena, 2017). However, due to some deficiencies of NMT systems such as the limited vocabulary size, and meaningless translations, much research work has involved combining NMT and SMT to improve translation performance (Cho et al., 2014; He et al., 2016; Niehues et al., 2016; Wang et al., 2017).

HMT is a strategy that combines different types of translation systems and fully takes advantage of the strengths of each system to improve translation performance. A typical example of HMT involves a rule-based system's predictable and consistent translations with an SMT system used for post-processing to further improve translation quality (Groves and Way, 2005; Paul et al., 2005; Groves and Way, 2006; Chen et al., 2007; Sánchez-Martínez et al., 2007; Enache et al., 2012; Li et al., 2015).[1] NMT is a new MT paradigm which can produce highly

[1] A huge amount of work has been done on this topic in the past decades, so we only list a representative sample here.

fluent translations. However, NMT sometimes generates translations that have a totally different meaning compared to the source sentences, which testifies to its strong language modeling but weak translation modeling capabilities. By contrast, PB-SMT is good at reflecting the adequacy of source sentences by means of the 'hard word alignment'. Intuitively, if we take the fluent but wrong translations as another language and perform word alignment, then we can perhaps restore the original meaning of source sentences to some extent using SMT. Moreover, for any source-side out-of-vocabulary (OOV) words in NMT, if we can keep them in the translations, then we can fully utilise the advantage of SMT to translate them.

Some work has been done on combining the adequacy of SMT and fluency of NMT. To the best of our knowledge, the most similar work is the pre-translation framework proposed in Niehues et al. (2016). In their framework, the SMT system is first used to pre-translate the input and then an NMT system generates the final hypothesis using the pre-translation as input. However, in their experiments, the "SMT⇒NMT" framework without integrating source information did not beat the pure NMT system and was not able to combine the strengths of both systems.

In this paper, we propose an "NMT⇒SMT" hybrid strategy to utilise SMT and NMT by considering (i) that NMT systems significantly outperform SMT systems, so using a higher-quality system as a post-processing step may indeed improve the performance of a lower-quality MT system, but might be difficult to outperform the higher-quality system; (ii) that NMT is more sensitive to noisy data compared to SMT, so using pre-translated data to train NMT will cause translation performance to deteriorate. Accordingly, the NMT system trained on the pre-translated data will not be able to correct errors from the SMT system. Experiments conducted on Japanese→English and Chinese→English demonstrate that our proposed "NMT⇒SMT" hybrid strategy can alleviate the above problems and further improve translation quality compared to pure NMT systems. The main contributions of this work include:

- We re-implement the "SMT⇒NMT" strategy on two different language pairs and four directions, namely Japanese↔English and Chinese↔English. Results show that this framework indeed cannot outperform the baseline NMT system.
- We propose an "NMT⇒SMT" hybrid framework that can better combine SMT and NMT by using their different strengths.
- We examine the effectiveness of the proposed framework on different NMT systems, namely the single NMT, factored NMT and ensemble NMT systems.

The rest of the paper is organised as follows. In Section 2, related work to the proposed neural hybrid MT framework is introduced. Section 3 describes the attentional encoder–decoder framework for NMT, and Section 4 introduces factored NMT and our proposed input features for NMT. In Section 5, we detail our proposed neural hybrid MT framework. In Section 6, we report the results of two sets of experiments on Chinese–English and Japanese–English tasks. Then a qualitative analysis is carried out, and some examples for comparing different systems are also illustrated in this section. Section 7 concludes and gives avenues for future work.

2 Related Work

The combination of NMT and SMT can be roughly categorised into three categories:

- NMT in post-processing: in this scenario, translations from SMT can be post-processed using NMT. For example, using NMT or neural networks to re-rank the outputs from SMT (Zhao et al., 2014; Lee et al., 2015; Neubig et al., 2015; Ding et al., 2016; Farajian et al., 2016), or using pre-translated results from SMT to build an NMT system (Niehues et al., 2016).

- Integrating SMT into NMT: in this scenario, SMT is used to guide translation in NMT, e.g. incorporating the translation model or language model into the decoding process of NMT (He et al., 2016; Wang et al., 2017).

- Integrating NMT into SMT: in this category, it is essentially integrating neural network-based features into SMT, such as the neural reordering model, neural language model, neural semantic model etc., e.g. Cho et al. (2014); Li et al. (2014); Passban et al. (2015).

In terms of the second category, He et al. (2016) incorporate SMT features, such as a translation model and an *n*-gram language model, with the NMT model under the log-linear framework. Their experiments show that the proposed method significantly improves translation quality of the baseline NMT system on Chinese→English translation tasks.

Wang et al. (2017) propose to incorporate an SMT model into the NMT framework in which at each decoding step, SMT offers additional recommendations of generated words based on the decoding information from NMT, and then an auxiliary classifier is employed to score the SMT recommendations and a gating function is used to combine the SMT recommendations with NMT generations, both of which are jointly trained within the NMT architecture in an end-to-end manner. Experimental results on Chinese-to-English translation show that the proposed approach achieves significant and consistent improvements over state-of-the-art NMT and SMT systems.

The proposed hybrid framework in this paper can be defined as a novel fourth category where we use SMT to post-process translations from NMT, which is completely different from the "SMT⇒NMT" framework in Niehues et al. (2016). In their framework, the SMT system is used to pre-translate the input and then an NMT system generates the final hypothesis using the pre-translation. In their experiments, the basic pre-translation system did not beat the NMT system either on natural order or pre-reordered data. By concatenating the source-side sentences with pre-translations as input to NMT, the final translation performance outperformed the baseline NMT system. From their results, we can see that the framework still needs the source information, and it is difficult to tell whether the improvements are mainly contributed by the pre-translation or source information.

3 Neural Machine Translation

The basic principle of an NMT system is that it can map a source-side sentence $\mathbf{x} = (x_1, \ldots, x_m)$ to a target sentence $\mathbf{y} = (y_1, \ldots, y_n)$ in a continuous vector space, where all sentences are assumed to terminate with a special "end-of-sentence" token $< eos >$. Conceptually, an NMT system employs neural networks to solve the conditional distributions in (1):

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(y_i|y_{<i}, x_{\leq m}) \tag{1}$$

We utilise the NMT architecture in Bahdanau et al. (2015), which is implemented as an attentional encoder-decoder network with recurrent neural networks (RNN).

In this framework, the encoder is a bidirectional neural network (Sutskever et al., 2014) with gated recurrent units (Cho et al., 2014) where a source-side sequence $\mathbf{x}$ is converted to a one-hot vector and fed in as the input, and then a forward sequence of hidden states $(\overrightarrow{h}_1, \ldots, \overrightarrow{h}_m)$ and a backward sequence of hidden states $(\overleftarrow{h}_1, \ldots, \overleftarrow{h}_m)$ are calculated and concatenated to form the annotation vector h_j. The decoder is also an RNN that predicts a target sequence $\mathbf{y}$ word by word where each word y_i is generated conditioned on the decoder hidden state s_i, the previous target word y_{i-1}, and the source-side context vector c_i, as in (2):

$$p(y_i|y_{<i}, \mathbf{x}) = g(y_{i-1}, s_i, c_i) \tag{2}$$

where g is the activation function that outputs the probability of y_i, and c_i is calculated as a weighted sum of the annotations h_j. The weight α_{ij} is computed as in (3):

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{m} exp(e_{ik})} \tag{3}$$

where

$$e_{ij} = a(s_{i-1}, h_j)$$

is an alignment model which models the probability that the inputs around position j are aligned to the output at position i. The alignment model is a single-layer feedforward neural network that is learned jointly through backpropagation.

4 Factored NMT Using Linguistic Features

Factored NMT, introduced in Sennrich and Haddow (2016), represents the encoder input as a combination of features as in (4):

$$\overrightarrow{h}_j = g(\overrightarrow{W}(\Big\|_{k=1}^{|F|} E_k x_{jk}) + \overrightarrow{U}\overrightarrow{h}_{j-1}) \tag{4}$$

where $\|$ is the vector concatenation, $E_k \in \mathbb{R}^{m_k \times K_k}$ are the feature-embedding matrices, with $\sum_{k=1}^{|F|} m_k = m$, K_k is the vocabulary size of the k_{th} feature, and $|F|$ is the number of features in the feature set F (Sennrich and Haddow, 2016).

In factored NMT, the features can be any form of knowledge which might be useful to NMT systems, such as POS tags, lemmas, morphological features and dependency labels as used in Sennrich and Haddow (2016). In our work, besides POS tags, we also use a new feature – word class (WoC) – in the NMT system. We define "POS+WoC" as pre-reordering features because they are used for pre-reordering source-side sentences in SMT (Neubig et al., 2012; Nakagawa, 2015).

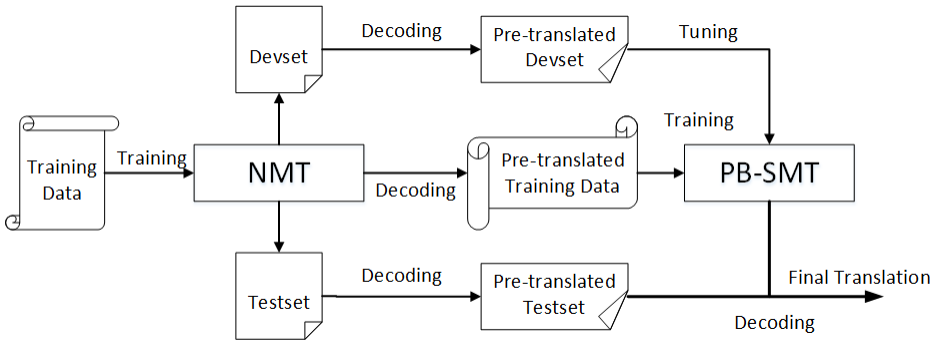


Figure 1: The Cascaded framework for neural HMT

5 Cascaded Hybrid Machine Translation

NMT can produce more fluent translations than SMT. However, NMT often produces some meaningless translations, i.e. the translation is totally different from the original meaning of the source sentence (Toral and Sánchez-Cartagena, 2017), or repeatedly translates some source words while mistakenly ignoring other words (Tu et al., 2017). We infer that this problem is due to the lack of an explicit translation model in NMT and the whole framework of NMT is regarded as a language model.

Although the soft-attention mechanism is helpful to guide the prediction of target words using the source information, and a reconstructor in NMT can manage to reconstruct the input source sentence from the hidden layer of the output target sentence to avoid the duplicate translation of source words (Tu et al., 2017), it still cannot explicitly and fully use the source information as in SMT. Thus, if we regard the translation from NMT as another 'language', and use SMT to perform word alignment and build a translation model, we might alleviate the meaningless and duplicate translations to some extent.[2]

Therefore, we propose an "NMT⇒SMT" framework to combine NMT and SMT as a multi-engine hybrid MT system as illustrated in Figure 1. In this pipeline, the first step is to train an NMT system using the initial training data, and then translate the training data, development set (devset) and test set (testset) into pre-translations; the second step is to use the pre-translated training data to build a target–target SMT system and tune the parameters using the pre-translated devset; the last step is to use the tuned SMT system to decode the pre-translated test set and produce the final output.

During pre-translation using NMT to translate the training data, devset and testset, we allow NMT to generate the 'UNK' token if an OOV occurs in the source sentence. Then, we propose a very simple but effective method to replace the "UNK" token in the translation by the corresponding source word. The method is shown in Algorithm 1.

Algorithm 1 Replacing UNK by source words

Require: A source sentence f_1^l, the translation e_1^m with UNK tokens, and the limited source-side vocabulary V for NMT.

```
source_position = 1
for i = 1 to m do
  if e_i == UNK then
    for j =source_position to l do
      if f_j not in V then
        e_i = f_j
        source_position = j + 1
        break
      end if
    end for
  end if
end for
```

The mechanism of Algorithm 1 is different from that in Jean et al. (2015) where they use the soft word-alignment information from the NMT system to guide the substitution process. However, generating the word-alignment information from NMT is quite time-consuming, especially for the translation of the training data. Therefore, our algorithm simply traverses the

[2]This is an open question. Intuitively, this depends on what word alignments are learned and what phrase pairs are extracted.

translation and its corresponding source sentence. Specifically, when we encounter the 'UNK' token, we will look up the source words in order in the NMT source-side vocabulary. If the source word does not exist in the vocabulary, then we replace the 'UNK' with this source word. We repeat this process to the end of the translation sentence. There might exist the wrong replacement due to different word order between the source sentence and the target sentence.[3] However, we believe that the SMT system will use its local reordering capability to reorder the OOVs to some extent and translate them.

Finally, different from using a back-off dictionary to post-process these unknown words in Luong et al. (2015b), we employ an SMT system to translate by considering more context.

6 Experiments

As Japanese and Chinese languages differ drastically from English in terms of word order and grammatical structure, we select Japanese–English and Chinese–English translations[4] to verify the proposed framework.

We also re-implement the "SMT⇒NMT" pipeline proposed in Niehues et al. (2016) as a comparison with our proposed framework. Therefore, two sets of experiments are set up as follows:

- "SMT⇒NMT": four translation directions (JP↔EN and ZH↔EN) are evaluated on natural-order and pre-reordered data. We employ the top-down BTG-based pre-reordering method to reorder source-side sentences (Nakagawa, 2015).
- "NMT⇒SMT": we test our proposed framework by integrating different types of NMT systems on JP→EN and ZH→EN tasks.

In the following sections, we report our experimental setup and results in terms of these two experiments.

6.1 Experimental Settings

For JP–EN translation tasks, the training data is the first part (train-1) of the JP–EN Scientific Paper Abstract Corpus (ASPEC-JE) that contains 1M sentence pairs, the development/validation set contains 1,790 sentence pairs, and the test set contains 1,812 sentence pairs (Nakazawa et al., 2016). There is only one reference for each source-side sentence in the validation and test sets.

For ZH–EN tasks, we use 1.4M sentence pairs extracted from LDC ZH–EN corpora as the training data, and NIST 2004 current set as the development/validation set that contains 1,597 sentences, and NIST 2005 current set as the test set that contains 1,082 sentences. There are four references for each Chinese sentence and there is only one reference for each English sentence in the validation and test sets. For the EN→ZH direction, we use the first reference out of four references for Chinese as the input (English).

For factored NMT, we use POS tags and word class as input features, which are obtained as follows:

- POS tag: the Japanese data are segmented and tagged using KyTea (Neubig et al., 2011), and the Chinese data are segmented and tagged using the ICTCLAS toolkit (Zhang et al., 2003).
- Word Class (WoC): the word classes of the training data are obtained using "mkcls" by setting the number of classes to 50. For an OOV word in the validation and test sets, we randomly allocate a class between (1, 50) to it.

[3] Noting that the number of OOVs in the source sentence is not always precisely the same as the number of UNKs in the translation of NMT. In our method, we take the minimum of these two numbers to replace the OOVs.

[4] In the rest of the paper, we use JP, ZH and EN to denote Japanese, Chinese and English, respectively.

Chinese and Japanese are not suitable for using the Byte Pair Encoding (BPE) method (Sennrich et al., 2016) to encode words as subword units, so we keep the words as translation units. We use Moses (Koehn et al., 2007) with default settings as the standard PB-SMT system, and use KenLM (Heafield et al., 2013) to train a 5-gram language model. We use Nematus (Sennrich et al., 2017) as the NMT system, and set minibatches of size 80, a maximum sentence length of 60, word embeddings of size 600, and hidden layers of size 1024. The vocabulary size for input and output is set to 45K. The models are trained with the Adadelta optimizer (Zeiler, 2012), reshuffling the training corpus between epochs. We validate the model every 5,000 minibatches via BLEU (Papineni et al., 2002) scores on the validation set and save the model every 30,000 iterations.

As in Sennrich and Haddow (2016), for factored NMT systems, in order to ensure that performance improvements are not simply due to an increase in the number of model parameters, we keep the total size of the embedding layer fixed to 600. Tables 1 and 2 show the vocabulary size and embedding size for pre-reordering features and the word as the input for the JP→EN and ZH→EN systems, respectively.

	Vocab. Size		Embedding Size	
Feature	Corpus	Model	All	Single
POS	21	21	10	10
WoC	51	51	10	10
Word	161,390	45K	580	590

Table 1: Vocabulary size, and size of embedding layer of pre-reordering features and words for JP→EN

	Vocab. Size		Embedding Size	
Feature	ZH	Model	All	Single
POS	36	36	10	10
WoC	51	51	10	10
Word	185,029	45K	580	590

Table 2: Vocabulary size, and size of embedding layer of pre-reordering features and words for ZH→EN.

In Tables 1 and 2, the columns from left to right under "Vocab. Size" indicate the vocabulary size of each feature. For example, "21" for "Corpus" indicates that there are a total of 21 POS tags in our Japanese corpus, and "21" for "Model" indicates that the vocabulary size of POS tags configured in the NMT model is 21. The column "All" indicates the embedding size of a feature when it combines with all other features, and "Single" indicates the embedding size of a feature only combining with "Word". In all NMT systems, the total embedding size is fixed to 600. Therefore, "590" indicates that for each single feature, the word embedding size for "Word" is obtained by $[600 - embedding\_size(feature) = 600 - 10 = 590]$.

The NMT system with the best BLEU score is selected as our baseline, and in terms of the ensemble NMT system, we use the last 5 models. The beam size for all NMT systems is set to 12.

We only employ the pre-translated training data and devset from the baseline NMT system to train and tune the SMT engine. Then the tuned SMT system is employed to re-decode the pre-translated test set using the baseline NMT, factored NMT and ensemble NMT systems, respectively.

	JP→EN				EN→JP			
	Non-reordered		Pre-reordered		Non-reordered		Pre-reordered	
SYS	Validation	Test	Validation	Test	Validation	Test	Validation	Test
SMT	18.25	17.64	**21.79***	**21.71***	27.03	26.32	**33.67***	**33.75***
NMT	**24.16***	**24.55***	20.42	21.43	**35.25***	**35.23***	32.75	32.98
SMT⇒NMT	18.01	17.83	20.39	20.91	27.64	27.57	33.23	33.43

Table 3: Results on JP–EN SMT⇒NMT experiments. "*" indicates translation performance is significantly better.

	ZH→EN				EN→ZH			
	Non-reordered		Pre-reordered		Non-reordered		Pre-reordered	
SYS	Validation	Test	Validation	Test	Validation	Test	Validation	Test
SMT	33.13	29.24	**34.63***	**30.59***	14.50	12.77	**16.12***	**13.77***
NMT	**35.49***	**31.76***	33.95	30.23	**15.97***	**15.62***	14.14	13.53
SMT⇒NMT	32.87	28.86	33.84	29.69	14.51	12.94	15.45	13.36

Table 4: Results on ZH–EN SMT⇒NMT experiments

All results are reported by case-insensitive BLEU scores and statistical significance is calculated via a bootstrap resampling significance test (Koehn, 2004).

6.2 Results and Analysis on SMT⇒NMT

Tables 3 and 4 show the results for JP↔EN and ZH↔EN with and without pre-reordered data, respectively. The baseline system is a standard PB-SMT system trained on non-reordered and pre-reordered data, respectively. "NMT" indicates the baseline NMT system as described in Section 6.1.

From Table 3, we can see that:

- Non-reordered task: all "SMT⇒NMT" systems on JP→EN and EN→JP are significantly worse than the baseline NMT systems. Except for the validation set of JP→EN, all other "SMT⇒NMT" systems on JP→EN and EN→JP outperform the baseline SMT systems.

- Pre-reordered task: the "SMT⇒NMT" system is worse than both the pre-reordered NMT system and pre-reordered SMT system on JP→EN, while it is better than the pre-reordered NMT system on EN→JP.

For ZH↔EN tasks, the "SMT⇒NMT" system performs worse relative to JP↔EN tasks, i.e. almost all "SMT⇒NMT" systems did not beat the NMT and SMT systems.

The observations from these experiments show that:

- if the translation quality of NMT is better than that of SMT, then using NMT as a post-processing module without integrating source-side information to re-decode translations from SMT cannot further improve translation performance.

- the pre-reordering in the source-side sentences is indeed helpful to SMT, while it hurts the performance of NMT.

- we need a better pipeline to combine NMT and SMT without using the source-side information.

6.3 Results and Analysis on NMT⇒SMT

Results on the proposed NMT⇒SMT model are shown in Table 5, where "NMT⇒SMT-B" indicates that the "NMT⇒SMT" pipeline re-decodes the translations of the baseline NMT system, "NMT⇒SMT-F" indicates that the "NMT⇒SMT" pipeline re-decodes the translations of the factored NMT system, and "NMT⇒SMT-E" indicates that the "NMT⇒SMT" system re-decodes the translations of the ensemble NMT system.[5]

	JP→EN		ZH→EN	
SYS	Validation	Test	Validation	Test
SMT	18.25	17.64	33.13	29.24
NMT	24.16	24.55	35.49	31.76
NMT⇒SMT-B	25.33*	25.66*	36.58*	32.38*
factored NMT	25.08	25.17	37.42	33.15
NMT⇒SMT-F	25.94*	26.08*	37.69*	33.39*
ensemble NMT	26.24	26.37	39.10	35.69
NMT⇒SMT-E	26.80*	26.93*	39.53*	35.98*

Table 5: Results of SMT⇒NMT experiments on JP→EN and ZH→EN. "*" indicates translation performance is significantly better.

We observe that:

- JP→EN: the NMT⇒SMT-B improves translation performance by 1.17 BLEU points and 1.11 BLEU points on validation and test sets, respectively, compared to the baseline NMT system. The NMT⇒SMT-F improves by 0.86 BLEU points and 0.91 BLEU points on the validation and test sets, respectively, compared to the factored NMT system. The NMT⇒SMT-E improves by 0.56 BLEU points and 0.56 BLEU points on the validation and test sets, respectively, compared to the ensemble NMT system, and improves by **2.64** BLEU points and **2.38** BLEU points on the validation and test sets, respectively, compared to the baseline NMT system. All improvements are significantly better.

- ZH→EN: the NMT⇒SMT-B improves translation performance by 1.09 BLEU points and 0.62 BLEU points on validation and test sets, respectively, compared to the baseline NMT system. The NMT⇒SMT-F improves by 0.27 BLEU points and 0.24 BLEU points on the validation and test sets, respectively, compared to the factored NMT system. The NMT⇒SMT-E improves by 0.43 BLEU points and 0.29 BLEU points on the validation and test sets, respectively, compared to the ensemble NMT system, and improves by **4.04** BLEU points and **4.22** BLEU points on the validation and test sets, respectively, compared to the baseline NMT system. All improvements are significantly better.

The results show that:

- Our proposed neural hybrid MT pipeline is more effective and feasible than the SMT⇒NMT pipeline. In Niehues et al. (2016), the SMT⇒NMT pipeline only works when integrating the source information into NMT. However, it increases the computational complexity by concatenating the pre-translated and source sentences as input to NMT.

[5]In current experiments, we only ensemble the baseline NMT systems. In future, we also plan to ensemble the factored NMT systems to verify the HMT performance.

- Our proposed NMT⇒SMT framework only uses source-side information once, i.e. at the stage of NMT training, while at the stage of post-processing, we only use the pre-translations without the source information (except OOVs), which keeps the framework simpler than the SMT⇒NMT framework.
- For different types of NMT systems, the proposed pipeline can significantly further improve translation performance, and the pre-translated SMT system is only trained using translations from the baseline NMT system. We would expect further improvements if we use the translations from the factored NMT or ensemble NMT models to train the SMT engine.

From the analysis on the results, we found that:

- OOVs rate in the test set is significantly decreased in the proposed framework, i.e. the post-processing SMT system can translate some of the OOVs appearing in the test set due to its larger vocabulary. For example, in the Chinese test set, the OOVs rate for NMT system is 4.62%. In the final result of the proposed framework, the OOVs rate is reduced to 2.36%.
- The improvement of translation performance is also attributed to the reordering and correction of phrases. We will carry out human evaluation and look into more details in future.

6.4 Examples

To further analyse the proposed NMT⇒SMT framework, Table 6 shows two examples produced from the baseline NMT system and the corresponding NMT⇒SMT from the JP→EN and ZH→EN tasks, respectively. The first example in Table 6 shows that the SMT system has the capability of making local translations more fluent. We can see that the NMT⇒SMT-B changes the phrase "the hydrogen bond network" in NMT to "hydrogen bond networks" which exactly matches the reference. "NMT-OOV" in the second example indicates the pre-translations after tracking the "UNK" symbols and replacing them by source-side OOVs. This example shows the capability of SMT to make the translation more adequate by subsequently translating the OOV.

Reference:	next , the change of **hydrogen bond networks** which was a basis of the **motion** of the water was explained .
NMT:	next , the change of **the hydrogen bond network** which was a basis of the movement of the water was explained .
NMT⇒SMT-B:	next , the change of **hydrogen bond networks** which was a basis of the movement of the water was explained .
Reference:	**barratt** said : " we have not achieved further information . "
NMT:	**UNK** said : " we have yet to get any results . "
NMT-OOV:	巴拉特 said : " we have yet to get any results . "
NMT⇒SMT-B:	**barratt** said : " we have yet to get any results . "

Table 6: Examples

7 Conclusion

In this paper we propose a cascaded hybrid framework (NMT⇒SMT) to combine NMT and SMT to improve translation performance. More specifically, we first employ a trained NMT system to pre-translate the training data, and then train an SMT system using the pre-translated

data. Finally, the tuned target–target SMT system is utilised to re-decode the pre-translated test set and produce the final results. We compare the proposed NMT⇒SMT pipeline with the SMT⇒NMT pipeline on JP–EN and ZH–EN tasks, and show that our framework is more effective than SMT⇒NMT, resulting in improvements on the test set of 2.38 BLEU points and 4.22 BLEU points on JP→EN and ZH→EN, respectively, compared to the baseline NMT system.

As to future work, we expect more experiments on different language pairs and larger-scale data sets to verify the proposed framework, and we will explore better combination of NMT and SMT to further improve translation quality. Additionally, we also want to verify the HMT framework without replacing OOVs in the NMT outputs.

Acknowledgements

We would like to thank the reviewers for their valuable and constructive comments. This research is supported by the ADAPT Centre for Digital Content Technology, funded under the SFI Research Centres Programme (Grant 13/RC/2106), and by SFI Industry Fellowship Programme 2016 (Grant 16/IFB/4490).

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, pages 1–15, San Diego, USA.

Bentivogli, L., Bisazza, A., Cettolo, M., and Federico, M. (2016). Neural versus phrase-based machine translation quality: A case study. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 257–267, Austin, Texas, USA.

Chen, Y., Eisele, A., Federmann, C., Hasler, E., Jellinghaus, M., and Theison, S. (2007). Multi-engine machine translation with an open-source decoder for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 193–196, Prague, Czech Republic.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 1724–1734, Doha, Qatar.

Ding, S., Duh, K., Khayrallah, H., Koehn, P., and Post, M. (2016). The JHU machine translation systems for WMT 2016. In *Proceedings of the Conference on Statistical Machine Translation*, Berlin, Germany.

Enache, R., España-Bonet, C., Ranta, A., and Màrquez, L. (2012). A hybrid system for patent translation. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, pages 269–276, Trento, Italy.

Farajian, M. A., Chatterjee, R., Conforti, C., Jalalvand, S., Balaraman, V., Gangi, M. A. D., Ataman, D., Turchi, M., Negri, M., and Federico, M. (2016). FBK's neural machine translation systems for IWSLT 2016. In *Proceedings of the 13th Workshop on Spoken Language Translation*, pages 8–15, Seattle, USA.

Groves, D. and Way, A. (2005). Hybrid example-based SMT: the best of both worlds? In *Proceedings of the Workshop on Building and Using Parallel Texts – Data-driven machine translation and beyond*, pages 183–190, Ann Arbor, USA.

Groves, D. and Way, A. (2006). Hybridity in MT: experiments on the Europarl corpus. In *Proceedings of the 11th Annual Conference of the European Association for Machine Translation*, pages 115–124, Oslo, Norway.

He, W., He, Z., Wu, H., and Wang, H. (2016). Improved neural machine translation with SMT features. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 151–157, Phoenix, Arizona, USA.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (2013). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria.

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2015). On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1–10, Beijing, China.

Junczys-Dowmunt, M., Dwojak, T., and Hoang, H. (2016). Is neural machine translation ready for deployment? A case study on 30 translation directions. In *Proceedings of the International Workshop on Spoken Language Translation*, Tokyo, Japan.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 388–395, Barcelona, Spain.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, pages 177–180, Prague, Czech Republic.

Lee, H.-G., Lee, J., Kim, J.-S., and Lee, C.-K. (2015). NAVER machine translation system for wat 2015. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 69–73, Kyoto, Japan.

Li, H., Zhao, K., Hu, R., Zhu, Y., and Jin, Y. (2015). A hybrid system for Chinese-English patent machine translation. In *Proceedings of MT Summit XV: Sixth Workshop on Patent and Scientific Literature Translation (PSLT6)*, pages 52–67, Miami, USA.

Li, P., Liu, Y., Sun, M., Izuha, T., and Zhang, D. (2014). A neural reordering model for phrase-based translation. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages 1897–1907, Dublin, Ireland.

Luong, T., Pham, H., and Manning, C. D. (2015a). Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.

Luong, T., Sutskever, I., Le, Q., Vinyals, O., , and Zaremba, W. (2015b). Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 11–19, Beijing, China.

Nakagawa, T. (2015). Efficient top-down BTG parsing for machine translation preordering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference of the Asian Federation of Natural Language Processing*, pages 208–218, Beijing, China.

Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S., and Isahara, H. (2016). ASPEC: Asian Scientific Paper Excerpt Corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, Portorož, Slovenia.

Neubig, G., Morishita, M., and Nakamura, S. (2015). Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 35–41, Kyoto, Japan.

Neubig, G., Nakata, Y., and Mori, S. (2011). Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA.

Neubig, G., Watanabe, T., and Mori, S. (2012). Inducing a discriminative parser to optimize machine translation reordering. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning*, pages 843–853, Jeju Island, Korea.

Niehues, J., Cho, E., Ha, T.-L., and Waibel, A. (2016). Pre-translation for neural machine translation. In *Proceedings of the COLING*, pages 1828–1836, Osaka, Japan.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, USA.

Passban, P., Hokamp, C., and Liu, Q. (2015). Bilingual distributed phrase representations for statistical machine translation. In *Proceedings of MT Summit XV*, pages 310–318, Miami, USA.

Paul, M., Doi, T., Hwang, Y., Imamura, K., Okuma, H., and Sumita, E. (2005). Nobody is perfect: ATR's hybrid approach to spoken language translation. In *Proceedings of the International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation*, Pittsburgh, USA.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Barone, A. V. M., Mokry, J., and Nădejde, M. (2017). Nematus: a toolkit for neural machine translation. In *arXiv:1703.04357*.

Sennrich, R. and Haddow, B. (2016). Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91, Berlin, Germany.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, Berlin, Germany.

Sutskever, I., Vinyals, O., , and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 2014 Neural Information Processing Systems*, pages 3104–3112, Montreal, Canada.

Sánchez-Martínez, F., Pérez-Ortiz, J. A., and Forcada, M. L. (2007). Integrating corpus-based and rule-based approaches in an open-source machine translation system. In *Proceedings of the METIS-II Workshop: New Approaches to Machine Translation*, Leuven, Belgium.

Toral, A. and Sánchez-Cartagena, V. M. (2017). A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain.

Tu, Z., Liu, Y., Shang, L., Liu, X., and Li, H. (2017). Neural machine translation with reconstruction. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, San Francisco, USA.

Wang, X., Lu, Z., Tu, Z., Li, H., Xiong, D., and Zhang, M. (2017). Neural machine translation advised by statistical machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, San Francisco, California, USA.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. In *arXiv:1609.08144*.

Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. In *CoRR, abs/1212.5701*.

Zhang, H., Yu, H., Xiong, D., and Liu, Q. (2003). HHMM-based chinese lexical analyzer ICTCLAS. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 184–187, Sapporo, Japan.

Zhao, Y., Huang, S., Chen, H., and Chen, J. (2014). Investigation on statistical machine translation with neural language models. In *Proceedings of Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 175–186.

Neural and Statistical Methods for Leveraging Meta-information in Machine Translation

Shahram Khadivi skhadivi@ebay.com
eBay Inc., Aachen, Germany

Patrick Wilken* patrick.wilken@rwth-aachen.de
RWTH Aachen University, Aachen, Germany

Leonard Dahlmann fdahlmann@ebay.com
Evgeny Matusov ematusov@ebay.com
eBay Inc., Aachen, Germany

Abstract

In this paper, we discuss different methods which use meta information and richer context that may accompany source language input to improve machine translation quality. We focus on category information of input text as meta information, but the proposed methods can be extended to all textual and non-textual meta information that might be available for the input text or automatically predicted using the text content. The main novelty of this work is to use state-of-the-art neural network methods to tackle this problem within a statistical machine translation (SMT) framework. We observe translation quality improvements up to 3% in terms of BLEU score in some text categories.

1 Introduction

Using larger context in machine translation to improve its quality, including selection of correct word meaning, has been a challenging task. Correct translation of polysemous words is vital to transfer important information from source sentence to the translation. To find the correct sense of polysemous words and phrases, usually only the context of the source sentence is available. Depending on the use case, the context can be extended to the surrounding sentences, or external signals about the text, like its topic or genre. Therefore, we can consider all methods that try to use a larger context for translation as methods that can help MT system select the right translation for polysemous words. In e-commerce, the problem of polysemous words is more severe. For example, incorrect literal translation of a brand name like Apple, Coach, Diesel, Affliction, Avenue, Cables To Go, Free People, etc. misleads a potential buyer. It also may create legal issues when e.g. a wrong translation directs buyers to a competitive brand name. In e-commerce MT scenarios, one of the main tasks is often the translation of the titles of the items offered for sale. Such titles are short, non-grammatical, and the local context of a given word is very variational. Since item title translation has a crucial importance in cross-border trade for e-commerce, we are trying to leverage meta data available for each item to deal with these irregularities. Items in e-commerce inventory are usually classified according to a hierarchical taxonomy. The hierarchy itself contains top-level categories (like *Clothing*, *Electronics*) with varying degrees of depth in each top category. Although such hierarchy is created based on

*Patrick Wilken has contributed to this work during his internship at eBay Inc.

business insights, it implicitly groups objects which can be described in semantically similar terms. Therefore, we expect less variation in word senses within a category, with ambiguity decreasing deeper in the tree. For instance, Apple is very likely to be a brand name, not a fruit in *Smartphones*. Therefore, item category information can potentially provide a strong signal to identify the meaning of a word. In this work, we focus on using more broad product categories of a particular e-commerce site, on the top level 1 (L_1, e.g. "Clothing and Shoes"), but also on levels 2 (L_2, e.g. "Women's Shoes") and 3 (L_3, e.g. "Women's Boots").

The main goal of this work is to modify the major state-of-the-art approaches that leverage larger context in MT to use (category) meta-information both in training and at runtime and check experimentally whether such approaches are able to better translate polysemous words in e-commerce data and improve MT quality both overall and in specific categories. Among others, we look at neural machine translation (NMT) models which are by now state-of-the-art and by definition use larger context during decoding, as in (Bahdanau et al., 2014). Since our goal is to enhance a real-time production phrase-based SMT system for title translation, and also to have a better understanding of the power of NMT as compared to SMT in using larger context, we use a bidirectional recurrent neural network (RNN) lexical model and also a feed-forward NN model as features in our SMT system. To the best of our knowledge, we were the first to modify these specific NMT models to use the embedding of sentence meta-information as an additional signal. Also, in this work we propose and test a simple generative category-specific word lexicon model.

The main challenges we encountered are data sparseness, data bias towards most frequently observed meaning of polysemous words, and absence of meta information for parts of training data. Nevertheless, significant improvements of MT quality could be achieved with some of the described methods on selected tasks.

In the next section, we describe major research works that employ larger context in MT. In Section 3, we describe the models we investigate in our work and the novel ways of integrating the category information into these models. Section 4 is devoted to experimental results which include automatic and human evaluation on an English-to-Italian e-commerce title translation task. The paper concludes with a summary and future works in Section 5.

2 Related Work

Phrase-based MT models have an intrinsic problem in using large(r) context and long range dependencies, since these models are bounded by phrase context. Therefore, many research publications target solving these well-known problems of phrase-based MT. Here, we focus on pioneering works that are most comparable to this work. One research line of using larger context in MT is to use sentence context for word sense disambiguation (Carpuat and Wu, 2007). Mauser et al. (2009) proposed the idea of employing a discriminative lexical model that uses sentence level information to predict a target word, this idea has been extended and enhanced in a recent work (Tamchyna et al., 2016). Tamchyna et al. (2016) have proposed to extend the discriminative model to also use the target prefix to predict the next target word, and also they enhance the model to calculate target word probabilities on-line during the search using a fast and efficient classification method based on the `Vowpal Wabbit`[1] toolkit. Devlin et al. (2014) proposed a neural joint lexical model that also employs a larger context around a given source word including previous generated target words, to predict the corresponding target word probability using a feed-forward neural network as the classifier.

Research works of Durrani et al. (2011) and (Guta et al., 2015) can also be considered as attempts to use larger context and long-range dependencies in phrase-based MT by modeling the dependencies between phrases.

[1] http://hunch.net/~vw/

The use of topic models in MT is another promising way to use larger context in lexical translation. The main idea is to use the topic models to infer the topic of the whole document/sentence, and then use it as a signal to the MT system to find the correct sense of the source word to translate (Eidelman et al., 2012). Hasler et al. have investigated different ways to use topic models in SMT (Hasler et al., 2012a, 2014b,a). They showed relatively small but consistent improvements when topic models are used inside SMT models.

This work is different from previous works in the following aspects:

- We use a bidirectional LSTM as an alternative to a feed-forward NN (Devlin et al., 2014) or maximum entropy-like classifiers (Tamchyna et al., 2016).
- In comprehensive experiments, we explore different methods to use additional meta-information in translation process.
- We conduct a case study on e-commerce data where meta-information and context seem to be more effective.
- We perform human evaluation to confirm and explain improvements of automatic MT quality measures.

In our evaluation on an e-commerce translation test set containing a mix of product categories, we observed moderate improvements using different approaches introduced in the literature. This is in agreement with previous works. For specific product categories, however, we obtained large and significant improvements with each method. These experiments confirm the benefit of using larger context and meta-information in translation. In addition, we found that the problem is far from being solved by the current approaches.

3 MT Models Leveraging Meta-information

3.1 Sparse Lexical Features

The main idea is to bias a SMT system towards the vocabulary and style of the target domain that can be inferred from the latent topics of the source sentence (Eidelman et al., 2012). We employ sparse lexical features (Hasler et al., 2012b) and sparse topic features on top of common dense features in a SMT system. Sparse lexical features are tuples composed of a single source word and a single target word. These features can be also extended with another binary feature representing coexistence of a specific topic or text category in the source sentence. Topic information can be obtained from topic models trained on the source side of the bilingual training corpus along with other available in-domain monolingual data in the same language. The features with topic information are triggered by the topic of the source sentence, that is, for a particular source sentence to be translated, only the features that have been seen with the topic of that sentence will fire.

We can also add information like topics or text categories for each phrase pair in the phrase-table. This information can be inferred from each phrase pair independent of the context or it can be inferred from the sentence pairs from which a given phrase pair is extracted. Therefore, each phrase pair is augmented with its topics, i.e., a vector of membership values of the phrase pair to each topic. The topic model is trained on an appropriate monolingual data in the source language. Then, based on the source side of each phrase or sentence pair, the topics and their probabilities, which again form a vector, are inferred from the topic model. We can combine both types of features to create a SMT more sensitive to the context, similar to (Mathur et al., 2015). The idea of a topic vector can be extended to any other context vector, i.e., the vector is simply a phrase-pair-specific representation of the meta-information in a continuous vector space.

3.2 Feed-Forward neural translation model

A neural network model previously used as a feature in a PBMT system is the neural network joint model (NNJM), a feed forward architecture presented in (Devlin et al., 2014). Assuming the target sentence $E : e_1, \ldots, e_I$ and given the source sentence $F : f_1, \ldots, f_J$, NNJM predicts a target word e_i given a window $f_{b_i-w/2}^{b_i+w/2}$ of size $w+1$ around the corresponding source word and a target history e_{i-v}^{i-1} of length v. This context is represented in the model by the concatenation of word embeddings corresponding to the source and target words. Because of the dependence on target context, the model has to be evaluated during search for each translation hypothesis. Using a full softmax as output layer, this would be computationally prohibitive. Instead we use noise-contrastive estimation (NCE) as described in (Zoph et al., 2016). We rely on the self-normalizing property of NCE and do not perform a manual normalization of the network outputs. To further reduce evaluation time, the hidden layer contributions of all words in the vocabulary at all possible input positions are precomputed before search. This leads to a model which is fast enough to be evaluated during the search.

Due to the their flexibility, neural networks models can be easily augmented with additional inputs to integrate any kind of context information. In this work, we integrate the product category information into the feed-forward model by creating a one-hot category vector and appending it to the concatenation of the word embeddings, as shown in Figure 1. The additional input can be included in the precomputation of the hidden layer in a straightforward manner, because it only adds one more input that can be looked up just like the ones from the source words. Variants like using an embedding for the category or appending it to the hidden layer have also been investigated. However, we have not seen significant differences in terms of the MT quality.

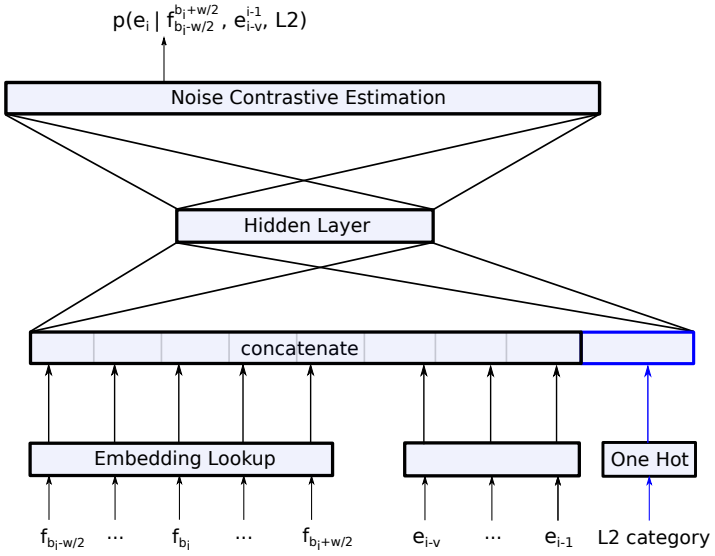


Figure 1: General architecture of neural network joint lexical model. Here, the L_2 product category is also shown as an optional input to the model.

3.3 Bidirectional RNN translation model

Recurrent neural network models can be efficiently integrated into the framework of a phrase-based machine translation system under some circumstances (Alkhouli et al., 2015). In this

work, we have adopted the encoder part of the bidirectional encoder-decoder machine translation architecture described in (Bahdanau et al., 2014). This bidirectional translation model (BTM) takes the whole source sentence as input and estimates the probability $p(e_i|f_1^I, b_i)$ of a translation of the source word at position b_i. This architecture is also similar to (Sundermeyer et al., 2014), but instead of the class-factored output layer we chose importance sampling (Jean et al., 2015) to train the model. Also, instead of introducing ϵ-tokens for unaligned words we obtain a unique b_i by applying Devlin's affiliation heuristic to the statistical word alignments (Devlin et al., 2014). Since the model does not depend on the target history the outputs for all combinations of e_i and b_i can be precomputed before the search. Thus, it is feasible to calculate the full softmax layer without approximations during the decoding.

Similar to the feed-forward model, we augment the BTM by adding category information as an additional input. In Figure 2, an example of the resulting network architecture is shown. The one-hot category vector is concatenated to the word embedding at each source position. Other methods like treating the category as a special word in front of the actual sentence or replacing the one-hot vector with a category embedding have also been investigated, but their performance was worse or the same as with the architecture in Figure 2.

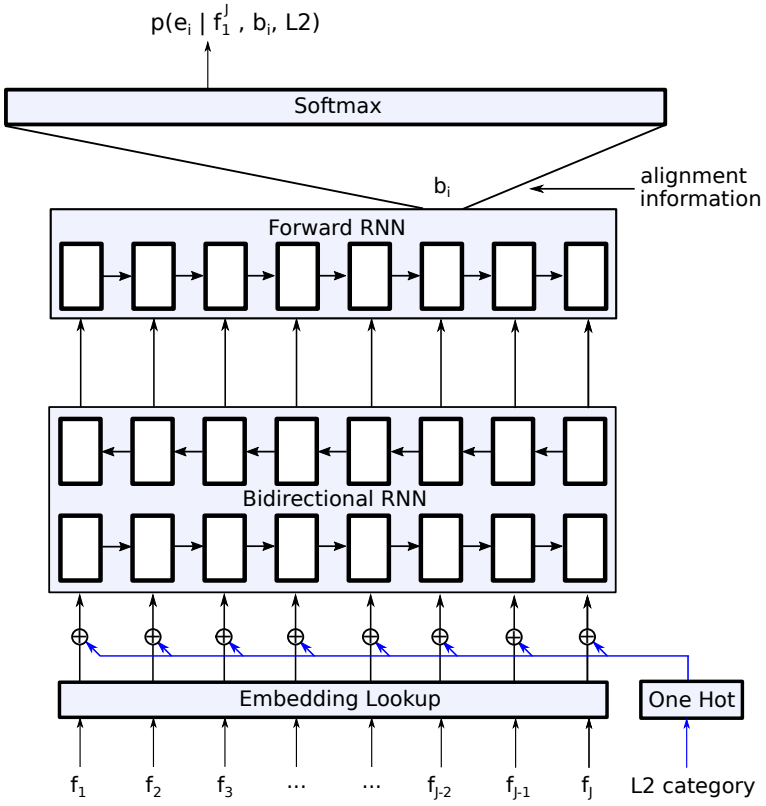


Figure 2: General architecture of bidirectional recurrent neural network lexical model. L_2 category is also shown as an optional input to the model.

3.4 Category-aware generative model

Here, the main idea is to introduce category L_2 as a hidden variable into the translation probability $p(e|f)$ of the target word e given the source word f originating from a title of category L_2:

$$\begin{aligned} p(e|f) &= \sum_{L_2} p(e, L_2|f) \\ &= \sum_{L_2} p(e|f) \cdot p(L_2|e, f) \\ &= \sum_{L_2} p(e|f) \cdot p(L_2|e) \end{aligned} \tag{1}$$

where $p(L_2|e)$ is the probability of a category L_2 given a target word e. This probabilistic function implicitly penalizes word and phrase translations that are rarely observed in L_2, and it favors target words frequently appearing in L_2, but not in other categories. We should note that $p(L_2|e)$ is different from category-specific language models of type $p(e|L_2)$. The main advantage of this model is the possibility to be trained on larger amounts of additional in-domain monolingual data in the target language that has category information.

4 Experiments

We conduct comprehensive experiments with various methods to see the impact of meta information on the translation of item titles in the e-commerce domain. We use two baseline phrase-based MT systems. The first one is based on the Moses toolkit (Koehn et al., 2007), and the second one is an in-house phrase-decoder (Matusov and Köprü, 2010), which is similar to the Moses decoder. In both systems, we use standard SMT features, including word-level and phrase-level translation probabilities, the distortion model, and an n-gram LM. Due to the nature of the item titles, we did not use any lexicalized reordering models in the MT system. The distortion limit was set to 6. On the target side, we built a trigram LM, which is optimum on this task, using KenLM (Heafield, 2011) trained with modified Kneser-Ney smoothing (Chen and Goodman, 1996). The LM is trained on the target side of bilingual data plus additional in-domain monolingual data composed of 60M words of item titles data. In addition, we have also used a 5-gram operation sequence model (OSM) (Durrani et al., 2011) and a 7-gram joint translation and reordering (JTR) model (Guta et al., 2015), which share the same idea and concept, in Moses-based and in-house systems, respectively. The feature weights are optimized using the k-best batch MIRA implementation provided in the Moses toolkit (Cherry and Foster, 2012). In the in-house decoder, the feature weights are tuned with minimum error rate training (MERT) Och (2003) on n-best lists of the development set. The MT quality is judged using the automatic case-insensitive BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) scores. Statistical significance tests were conducted using approximate randomization tests (Clark et al., 2011).

We have implemented the NNJM described in Section 3.2 and the BTM described in Section 3.3 using TensorFlow[2] The trained models are then exported to our in-house decoder using TensorFlow's C++ API. As the model is independent of the target history, we are able to precalculate scores for all phrase pairs that are selected in the phrase matching step for a given source sentence. This improves the runtime cost, as the model does not need to be queried during the decoding.

For training and evaluating feed-forward neural network models in Moses, we rely upon the Neural Probabilistic Language Model Toolkit (NPLM) (Vaswani et al., 2013). NPLM can

[2] www.tensorflow.org

be used to train both neural language models and joint models. We also integrate the models into our in-house decoder as language models.

		English	Italian
Train:	Sentences	10,231,392	
	Tokens	117 M	115 M
	Vocabulary	493 K	582 K
	Singletons	239 K	257 K
Dev.:	Sentences	910	910
	Tokens	10,818	11,159
	Vocabulary	4,422	4,481
	OOVs	321	310
Test	Sentences	910	910
	Tokens	10,814	11,241
	Vocabulary	4,487	4,532
	OOVs	337	321

Table 1: Corpus Statistics

We conduct experiments on an English-to-Italian e-commerce item titles translation task. The training data is composed of in-house data (item titles, descriptions, etc.) as in-domain corpus and also publicly available data that has been sampled according to the similarity to in-domain data. The corpus statistics are summarized in Table 1. The size of in-domain data is about 4.6% of the training data in terms of source side tokens. Before calculating the corpus statistics, we apply some usual text pre-processing including tokenization and replacement of numbers with a placeholder token; and also some domain dependent processing such as replacing product specification - e.g., `6S`, and `1080p` - with a general token.

The in-domain data has also some meta information that identifies the category of each sentence/segment. We denote this meta-information as L_2. In all experiments we define six categories, five selected L_2 categories plus one *other* category. Meta information for out-domain training data is inferred based on a state-of-the-art text classification algorithm trained on a big in-domain source monolingual data, for which the L_2 category is available. The Dev and Test sets also contain the category information, and they have two human reference translations.

In Table 2, the translation results are shown in terms of both BLEU score and TER. Moses baseline has a slightly better quality performance compared to in-house baseline system, seventh row. We report in-house results since NNJM and BTM models described respectively in Section 3.2 and Section 3.3, are implemented in this in-house system. We also report the results of a state-of-the-art baseline NMT system as described in (Chen et al., 2016). Based on Chen et al. (2016) and also our experiments on another language pair in the same task, there is room to improve this baseline using techniques like back-translation (Sennrich et al., 2015) and guided-alignment, but the performance of a single system (and even with ensembling technique) is lower than a strong phrase-based baseline. The lower performance is due to the nature of item titles data that are not appropriate for NMT approach. Item titles data are not grammatical, they are very irregular and like other user-generated data very noisy. Therefore, we confine to report a simple NMT baseline to show the characteristics of the task.

System in the second row is based on the work of Tamchyna et al. (2016), we have used the default features proposed in the original work: source bag of words, target bi-grams, source indicator and target indicator. We have also conducted two experiments to train the classifier model on in-domain data and on mixed domain data, we report the translation results if the

#	System	Test	
		BLEU [%]	TER [%]
1	Moses Baseline	37.4	45.7
2	+ Tamchyna et al. (2016)	37.4	45.9
3	+ Word-pair SF	37.6	45.9
4	+ Category SF	38.3	44.8
5	+ Category SF +Mathur et al. (2015)	37.5	45.4
6	+ NNJM	37.7	45.0
7	In-house Baseline	37.0	46.0
8	+ BTM	37.7	45.5
9	+ NNJM	37.5	45.5
10	Pure NMT baseline	28.0	54.9

Table 2: Experimental results: English-to-Italian item title translation task.

classifier model is trained only on in-domain data. In third and fourth experiments in Table 2, we use word-pair feature, and word-pair plus category information as sparse features (SF), respectively. We adopt the method presented in (Hasler et al., 2012b) in our case study, we replace topic models with predefined category information. We include the work of Mathur et al. (2015) in our experiments, with this difference that we use category information instead of topic models. As required by this model, we have augmented to each phrase of the phrase table a six-value normalized vector that represents the membership value of each phrase to each category. These membership values are simply normalized frequencies of categories in the parent sentence pairs of a phrase pair. Parent sentence pairs are those that include a given phrase pair. The next systems in the table are based on neural network models. In NNJM, we set a window of four source words. Since the translation of polysemous words is not directly dependent on the previous target words, and also, to make this model more comparable to BTM, we do not use target words in NNJM. In NNJM, input word embedding and output word embedding are set to 150 and 750, respectively. We use a single hidden layer in NNJM. Despite the same setting in sixth and ninth rows, the implementation of NNJM in the sixth row is based on NPLM toolkit in Moses, and implementation in ninth row is as described in Section 3.2 in TensorFlow toolkit. In BTM, we have used the embedding size of 620, RNN size of 1000 and GRU cells. The learning rate was set to 0.0002, decaying by 0.9 each epoch. The vocabulary is set to most frequent 100,000 words for both NNJM and BTM.

As shown in Table 2, the differences of MT systems are relatively small. Among other reasons, limited number of samples per each targeted categories in the test set could explain these small differences. The number of occurrences for each category are shown in the last row of Table 3, a similar distribution of categories exists for the development set.

In Table 3, we have also shown the detailed improvements achieved on five specific L_2 categories. Now, we observe much larger improvements in the selected categories. We observe the best performing system in Table 2 is not necessarily the best system for our specific goal that we embed corresponding additional information into the translation process. As shown in Table 3, in-house system plus NNJM has the most consistent improvements over its corresponding baseline. For a better comparison of results, you may use diagrams in Figure 3 and Figure 4. Please note that the results for all neural models are without category information so far. Although we expect improvements of MT quality on the particular categories by adding the category information as an additional signal into the neural models, we have not seen any signif-

Sys.	Cat. I		Cat. II		Cat. III		Cat. IV		Cat. V	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
Moses Baseline	58.5	27.3	36.2	44.2	35.8	47.2	29.6	51.7	34.1	48.2
+ Tamchyna et al. 2016	58.5	29.6	37.4	45.5	35.0	48.5	30.0	51.9	34.0	49.0
+ Word-pair SF	58.6	28.7	35.7	45.3	36.0	47.7	29.3	51.7	33.7	49.5
+ Category SF	60.6	27.1	36.2	43.9	35.8	47.7	30.1	50.2	34.0	48.7
+ Category SF + Mathur et al. 2015	59.3	28.0	36.8	42.8	38.0	46.9	29.6	50.7	34.3	48.2
+ NNJM	58.5	28.7	36.2	44.3	36.0	46.9	33.6	48.6	33.8	47.1
In-house Baseline	57.7	28.5	36.9	44.3	34.9	48.0	29.2	52.4	33.9	47.7
+ BTM	59.7	27.8	37.7	44.1	36.4	46.7	28.1	51.7	34.4	46.6
+ NNJM	59.9	27.8	37.8	43.4	36.6	46.4	32.2	50.0	35.1	46.6
# test sentences	33		62		27		32		30	

Table 3: Translation of English titles from five selected product categories into Italian (BLEU scores and TER in %).

icant improvements. We thought, we may need to send a stronger signal in the training process and therefore we have tried different ways to embed category information into the model, but the results in all cases were almost at the same level. These results are in contrast with the results reported in (Chen et al., 2016), they have shown some improvements using category information in an NMT architecture. This disagreement might be due to different architectures and also due to different baselines.

To investigate why category information does not contribute to MT quality improvements in our experiments, we have conducted some experiments to measure the perplexity on the training corpus when the BTM model is trained with and without L_2 information and we have observed a small increase in perplexity when we used the L_2 categories. We may discuss these results in two directions, first the category information are too sparse in our settings to be useful, and second, the category information has no more information over the text itself, especially when the text is processed globally in a neural network model.

We have also conducted an in-house human analysis of polysemous words to better understand the situation. We observe some examples where L_2 categories help to disambiguate the meaning of the words. At the same time, there are other polysemous words for which category information cannot help. For example, the sense of the word `Vans` can be identified if we know it is from `Motors` category meaning plural of `Van` or from `Clothing` category meaning a brand name. Another example is word `mixer` that has two different meaning in `Kitchen` and in `Music instruments` categories. However, there are cases that product categories will not help to disambiguate the meaning of the word, e.g., word `Ship` in category `Toys` may have at least two different meanings as a noun or as a verb. Therefore, L_1 or L_2 categories information might be not helpful for some polysemous words.

The system described in Section 3.4 uses an additional probability $p(L|e)$ for the input sentence category given a target word candidate. We have implemented it as an additional lexical model that assigned a score to each target phrase pair. Using this model with a tuned weight in the log-linear model combination did not result in significant improvement of automatic MT measures, the BLEU/TER scores remained basically the same as in-house baseline, line 7 of Table 2. However, this system was stable in the sense that its translations did not differ much from the baseline, and the observed changes predominantly affected content words. A manual analysis showed several examples where translations of polysemous words which were inappropriate for a certain product category were changed for the better. For instance, the Italian translation of the term `latte` used in an English title in the category `Bags and`

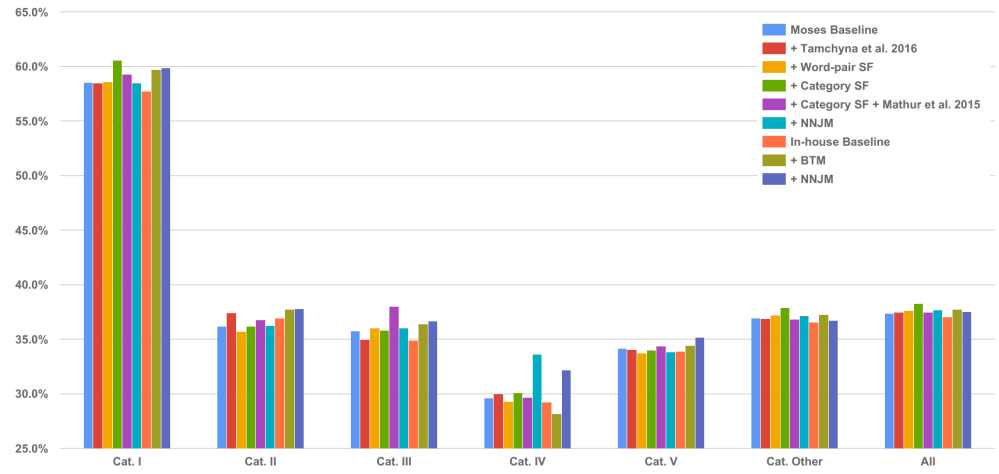


Figure 3: Detailed BLEU results.

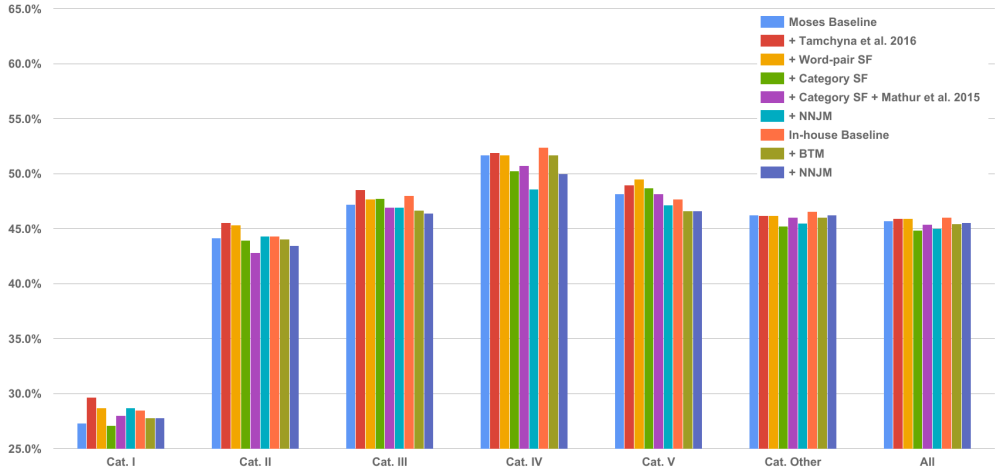


Figure 4: Detailed TER results.

`Accessories` as a bag's color was corrected from `latte macchiato` to `color crema`. Also, in the `Kitchen Appliances` category, the word `tamper` was incorrectly translated as `manomissione (fabrication)` by the baseline system. The system that uses the category-specific word lexicon $p(L|e)$ has correctly translated it as `pestello`.

In general, systems with the category-aware features, implicitly or explicitly, cause moderate improvements compared to a baseline system in terms of automatic MT error measures like BLEU. However, it can be argued that these automatic MT measures do not reflect the impact of polysemous words, since such words occur rarely as compared to all other words even if we consider all polysemous words, not only those which have a wrong translation in the baseline. A human evaluation focusing on translation of polysemous words can potentially justify the soundness of the employed method. For such an evaluation, we randomly selected 300 item titles if they had at least one of the polysemous words which have been identified as such by domain expert linguists. We asked human translators to answer five questions for each transla-

tion, the most important and relevant question was whether the identified polysemous word in the text was correctly translated or not. The human evaluation results show that although the proposed system cause moderate improvements - about the same as we observe in the above tables - one third of the polysemous words have not been translated correctly.

We attribute the weak performance of the presented models on polysemous words to the bias that we observe in the training data to the most frequent meanings of such words. For example, the translation of `Apple` as a brand name is by far more frequent in our training data than translation of this word as a common noun with the meaning of a fruit. Optimization for the BLEU score seems to additionally increase the bias, since the tuning set is in most cases also biased to the most frequent meaning of such words.

5 Conclusion

We employed several different ways to incorporate explicit meta-information or larger context to better translate polysemous words or improve MT quality in general. We explored existing state-of-the-art methods that can potentially help in this task or can accept another input as meta-information, e.g., (Hasler et al., 2012b; Mathur et al., 2015; Tamchyna et al., 2016; Devlin et al., 2014).

To better exploit the source-side topic/category labels, we introduced a bidirectional LSTM to encode the entire source sentence context to translate a word. We investigated different ways of incorporating meta-information in the encoder. In addition, we proposed a novel generative model that can leverage topic-labeled target monolingual data.

We conducted comprehensive experiments on different ways of using additional meta-information in translation process, including both the given human-labeled and the automatically predicted meta-information. Our case study was an e-commerce English-to-Italian translation task. We observed improvements up to 3% in terms of the BLEU score for some input text categories. Finally, we performed a human evaluation to confirm and explain improvements of automatic MT quality measures. We have realized that, although the observed improvements were reconfirmed by human evaluation, there were many polysemous words in the test set that were still not translated correctly. The take-home message of this research is that the problem of how to best use meta-information in MT for correct, in-topic translation of polysemous words and phrases is still far from being solved.

In the future, we aim to use other types of meta-information that may be better suited for disambiguating the meaning of polysemous words. For example, we plan to leverage automatically predicted domain-specific named-entity tags as meta-data for translation. Another area for future work is how to use meta-information more effectively, overcoming data sparseness and bias problems.

In the future, we also plan to adopt a hybrid NMT and SMT approach similar to (Dahlmann et al., 2017) to improve the translation of polysemous words in item title domain. In this way, we can benefit from long context coverage of NMT system to find a more appropriate translation based on the context for polysemous words, and also benefit from more control over the generation process of SMT system and its features like text override.

References

Alkhouli, T., Rietig, F., and Ney, H. (2015). Investigations on phrase-based decoding with recurrent neural network language and translation models. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, pages 294–303.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.

Carpuat, M. and Wu, D. (2007). Improving statistical machine translation using word sense disambiguation. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 61–72.

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chen, W., Matusov, E., Khadivi, S., and Peter, J.-T. (2016). Guided alignment training for topic-aware neural machine translation. *AMTA 2016, Vol.*, page 121.

Cherry, C. and Foster, G. F. (2012). Batch tuning strategies for statistical machine translation. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 427–436.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 176–181. Association for Computational Linguistics.

Dahlmann, L., Matusov, E., Petrushkov, P., and Khadivi, S. (2017). Neural machine translation leveraging phrase-based models in a hybrid search. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. M., and Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1370–1380.

Durrani, N., Schmid, H., and Fraser, A. M. (2011). A joint sequence translation model with integrated reordering. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1045–1054.

Eidelman, V., Boyd-Graber, J., and Resnik, P. (2012). Topic Models for Dynamic Translation Model Adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 115–119, Jeju Island, Korea. Association for Computational Linguistics.

Guta, A., Wuebker, J., Graca, M., Kim, Y., and Ney, H. (2015). Extended translation models in phrase-based decoding. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, pages 282–293.

Hasler, E., Blunsom, P., Koehn, P., and Haddow, B. (2014a). Dynamic Topic Adaptation for Phrase-based MT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–337, Gothenburg, Sweden. Association for Computational Linguistics.

Hasler, E., Haddow, B., and Koehn, P. (2012a). Sparse Lexicalised Features and Topic Adaptation for SMT. In *Proceedings of the Workshop on Spoken Language Translation (IWSLT)*, pages xxx–xxx.

Hasler, E., Haddow, B., and Koehn, P. (2012b). Sparse lexicalised features and topic adaptation for SMT. In *International Workshop on Spoken Language Translation, IWSLT 2012, Hong Kong, December 6-7, 2012*, pages 268–275.

Hasler, E., Haddow, B., and Koehn, P. (2014b). Combining domain and topic adaptation for SMT. In *Proceedings of the Eleventh Conference of the Association for Machine Translation in the Americas (AMTA)*, volume 1, pages 139–151.

Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2015). On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1–10.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Mathur, P., Federico, M., Köprü, S., Khadivi, S., and Sawaf, H. (2015). Topic adaptation for machine translation of e-commerce content. *Proceedings of MT Summit XV*, page 270.

Matusov, E. and Köprü, S. (2010). AppTek's APT Machine Translation System for IWSLT 2010. In Federico, M., Lane, I., Paul, M., and Yvon, F., editors, *International Workshop on Spoken Language Translation, IWSLT*, pages 29–36.

Mauser, A., Hasan, S., and Ney, H. (2009). Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 210–218.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2015). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

Sundermeyer, M., Alkhouli, T., Wuebker, J., and Ney, H. (2014). Translation modeling with bidirectional recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 14–25.

Tamchyna, A., Fraser, A. M., Bojar, O., and Junczys-Dowmunt, M. (2016). Target-side context for discriminative models in statistical machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Vaswani, A., Zhao, Y., Fossum, V., and Chiang, D. (2013). Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392, Seattle, Washington, USA. Association for Computational Linguistics.

Zoph, B., Vaswani, A., May, J., and Knight, K. (2016). Simple, fast noise-contrastive estimation for large RNN vocabularies. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1217–1222.

Translation Quality and Productivity: A Study on Rich Morphology Languages

Lucia Specia l.specia@sheffield.ac.uk
Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, UK

Kim Harris kim_harris@textform.com
text & form GmbH, 10179 Berlin, Germany

Frédéric Blain f.blain@sheffield.ac.uk
Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, UK

Aljoscha Burchardt aljoscha.burchardt@dfki.de
Viviven Macketanz vivien.macketanz@dfki.de
Language Technology Lab, DFKI, Alt-Moabit 91c, 10559 Berlin, Germany

Inguna Skadiņa inguna.skadina@tilde.lv
Tilde, Vienibas gatve 75a, Riga, LV 1004, Latvia

Matteo Negri negri@fbk.eu
Marco Turchi turchi@fbk.eu
Fondazione Bruno Kessler, Trento, 38100, Italy

Abstract

This paper introduces a unique large-scale machine translation dataset with various levels of human annotation combined with automatically recorded productivity features such as time and keystroke logging and manual scoring during the annotation process. The data was collected as part of the EU-funded QT21 project and comprises 20,000–45,000 sentences of industry-generated content with translation into English and three morphologically rich languages: English–German/Latvian/Czech and German–English, in either the information technology or life sciences domain. Altogether, the data consists of 176,476 tuples including a source sentence, the respective machine translation by a statistical system (additionally, by a neural system for two language pairs), a post-edited version of such translation by a native-speaking professional translator, an independently created reference translation, and information on post-editing: time, keystrokes, Likert scores, and annotator identifier. A subset of 2,000 sentences from this data per language pair and system type was also manually annotated with translation errors for deeper linguistic analysis. We describe the data collection process, provide a brief analysis of the resulting annotations and discuss the use of the data in quality estimation and automatic post-editing tasks.

1 Introduction

Data-driven approaches to machine translation (MT) rely largely on datasets of source sentences and their corresponding translations previously created by humans, so-called parallel corpora. MT systems, be they statistical or neural, are built in static fashion and (if at all) updated from time to time as more translations become available. With the popularisation of post-editing

(PE), a natural question is whether the corrected version of the MT output could be used in feedback loops to improve the current system via model retraining, model tuning or the addition of explicit model components. Additionally, by studying PE data, one can get insights on the errors made by the MT system to try and remedy them in different ways. PE data can also be used to build and benchmark metrics for the automatic evaluation of MT output, as well as quality estimation metrics and automatic PE systems.

To facilitate research in these and related areas, we have created a unique large-scale dataset with various levels of human annotation combined with automatically recorded productivity features. The data comprises 20,000–45,000 sentences of industry-generated content for English from or into three morphologically rich languages and was collected as part of the EU-funded QT21 project. The PE of all four language pairs was performed using a tool to record detailed process and product information at the sentence level during PE, including time, keystrokes, actual edits and Likert scores for the PE effort as given by the translator immediately after completion of the editing.

Most of the data was translated by a phrase-based statistical MT (PBMT) system. In addition, subsets of 15,000–20,000 sentences for EN–DE and EN–LV – respectively – were also translated using a neural MT (NMT) engine that was trained on exactly the same data used to train the original PBMT system. The PE of identical input data for both the PBMT and NMT systems facilitates large-scale direct comparisons between the actual output of these systems, as well as between process cues. For example, PE productivity can be calculated and compared using the time and keystroke information recorded during PE. The "preference" of translators can be compared through the scores given to the *perceived* quality of the output by such translators. A number of other comparative analyses and benchmarking in both research and industry scenarios become possible with this data.

Finally, a subset of 2,000 sentences was selected for each language pair and MT system type and manually annotated with word-level errors for deeper linguistic analysis. Both PE and error annotations were performed by professional translators.

While other datasets with PE data have been created in the past and also released for research purposes, these are limited in either their scale (e.g. see those used for the WMT13–14 shared tasks on quality estimation[1]), have been post-edited by non-professional translators (Wisniewski et al., 2013; Bojar et al., 2015), or make only the actual post-edits available, providing no additional information on the process and no explicit annotations. The most notable example of the latter is the Autodesk dataset (Zhechev, 2012). It contains sentences predominantly belonging to Autodesk software user manuals, covering 13 language pairs with English as the source language. The source sentence, its machine translation and its post-edit are provided. The translated sentences are produced by an MT system or are translation memory suggestions with a fuzzy match score larger than 75%.

In the remainder of this paper we first describe our data sources (Section 2) and the MT systems built (Section 3) to translate this data. We introduce the PE process and its results in Section 4, and the error annotation in Section 5. In Section 6 we present two uses of the dataset.

2 Data

The post-edited and annotated data described in this paper belongs to two specific domains: information technology (IT) and life sciences. These domains were chosen because of the high demand for this type of content in multiple languages due to its economic impact on businesses active on global markets where language is key. The use of this data in research can therefore play a significant role in building the necessary bridges between the constituencies most interested in achieving progress in the field of MT: research and industry.

[1] http://www.statmt.org/wmt14/

Language pair	# sentences	# source tokens	# target tokens	Domain	Data provider
EN–DE	80,874	1,322,775	1,312,975	IT	Adobe
EN–CS	81,352	1,332,654	1,175,463	IT	Adobe
EN–LV	231,028	3,713,803	3,168,740	Pharma	EMEA
DE–EN	193,637	3,120,482	3,228,761	Pharma	EMEA

Table 1: Domain-specific datasets: number of sentences and source and target tokens.

	Training data			
	EN–DE	EN–CS	EN–LV	DE–EN
# sentences	21,873	32,352	204,528	135,884
# source words	0.53	0.59	3.19	2.41

Table 2: Statistics on the in-domain training data. The number of words is reported in millions.

Four sets of parallel data in four language combinations (English– German/Latvian/Czech and German–English) were selected from the web. English Adobe software manuals translated into German and Czech were chosen for the IT domain, and a subset of the European Medicines Agency (EMEA) corpus was selected for the life sciences domain (which we also refer to as "pharma") to cover the English–Latvian and German–English language pairs.[2]

To create datasets that can satisfy different research needs and thus increase their usability, a set of criteria was applied to data selection and pre-processing. For English–German/Czech and German–English, sentences that did not end with a punctuation mark or contained less than three or more than 35 words were discarded, and duplicate sentences were removed. These strategies reduced the number of sentence pairs by approx. 45%. For English–Latvian, a part of parallel sentences were obtained by extracting textual sentences from PDF files in the EMEA repository. First, we used Adobe Acrobat v10 Professional to convert PDF files to HTML format, as this preserved most of the original document structure. Then we ran customised scripts to convert the HTML files to plain text and clean the data. The Microsoft Bilingual Sentence Aligner (Moore, 2002) was used for sentence alignment of the parallel plain text files. Duplicate sentence pairs and sentences with less than three or more than 35 words were removed. This sentence size filtering only marginally affected the size of the final corpus. The statistics of the final sets are reported in Table 1.

For each language pair, we selected a subset of data for annotation (see Table 5), and used the remaining sentence pairs as in-domain training data to build the MT systems (Section 3). This remaining data was split into training (see Table 2), development (2,000) and test (2,000) sets.

3 MT Engine Building

3.1 Training Data

A crucial aspect for creating a set of reliable post-edited sentences and error annotations is the availability of domain-adapted translations. This is necessary because a generic translation system is not able to correctly translate domain-specific terms or expressions, which would, in turn, cause translators to rewrite translations from scratch, rendering accurate error annotation

[2]The German–English dataset was created by taking the available English–German data and then inverting the language direction. This is not ideal; however, very little domain-specific data exists for under-resourced language pairs, including those whose source language is German.

	EN–DE		EN–CS		EN–LV		DE–EN	
	Parallel	Mono	Parallel	Mono	Parallel	Mono	Parallel	Mono
In-domain	7.2	-	-	-	0.181	-	2.09	2.35
Out-domain	12.7	-	50.34	51.46	-	-	-	-

Table 3: External resources collected to train the MT systems. The reported numbers represent millions of sentences.

impossible.

When building a domain-adapted MT system we rely on different external resources depending on the size of the in-domain data. For the language pairs for which there are less than 100,000 in-domain sentence pairs (*i.e.* EN–DE and EN–CS), a large collection of in- and out-of-domain monolingual and parallel corpora was gathered from the web, while for the remaining languages (EN–LV and DE–EN) only in-domain corpora were used. This process resulted in:

- EN–DE: Over 20 million generic and in-domain sentence pairs obtained by merging the datasets available in the OPUS (Tiedemann, 2012), TAUS, WMT and JRC [3] repositories (e.g. Europarl, CDEP, CommonCrawl, etc.);
- EN–CS: Over 51 million generic and in-domain sentence pairs available in the CzEng 1.6 dataset (Bojar et al., 2016b).[4] In addition, translating into a language with free word order suggests the use of a large collection (more than 50M sentences) of monolingual generic data obtained from the Translation task at WMT16;
- EN–LV: Over 385,000 parallel medical sentences from the EMEA corpus available in OPUS and the most recent documents from the EMEA website (years 2009-2014);
- DE–EN: Over 2 million in-domain sentence pairs collected from OPUS and the data released for the medical translation task at WMT14 (Bojar et al., 2014). These resources include MuchMore, PatTr, and the Wikipedia parallel titles. In addition to these parallel sentences, monolingual data (approx. 2 million) obtained from the medical translation task at WMT14.

A summary of the external resources used to train the MT system is shown in Table 3.

3.1.1 Data Selection

In MT literature, it has been shown that when large generic datasets and a small in-domain corpus exist, the use of data selection techniques can help improve translation quality (Eetemadi et al., 2015). To optimally leverage a domain-specific corpus, we used cross-entropy-based selection for monolingual data (Moore and Lewis, 2010), its extended version for bilingual texts proposed by Axelrod et al. (2011) and the latent-domain translation method (Cuong and Simaan, 2014).

Entropy-based method: Originally proposed by Gao and Zhang (2002), entropy-based approaches consist in computing the perplexity score of each sentence of a generic corpus against both an in-domain language model (LM) and an LM trained on the generic corpus. The sentences are then ranked according to the difference between their two perplexity scores. Once all of the generic sentences have been ranked, the size of the subset to extract is determined by minimising the perplexity of a development set against an LM trained on an increasing amount of the sorted corpus (e.g. 5%, 10%, ...). According to (Moore and Lewis, 2010), perplexity

[3] https://ec.europa.eu/jrc/en/language-technologies

[4] http://ufal.mff.cuni.cz/czeng/czeng16pre

decreases when less but more relevant data is used. We used the freely available open-source tool XenC (Rousseau, 2013).

Latent-domain translation method: This technique is able to give priors to different domains that comprise the generic data set. The goal is to estimate the probability of whether a sentence pair belongs to the in- or out-of-domain data, using in-domain corpus statistics as prior. The Expectation-Maximisation training algorithm is derived and used to estimate the out-of-domain models (given only in and mixed-domain data). This technique provides the selected data directly without the need to choose a cut-off point in the ranked list of sentence pairs.

Both methods were first tested on the EN–DE language pair, and the best performing method was applied to EN–CS. In our experiments, we used the data shown in Table 1 as in-domain and the concatenation of the data in Table 3 as out-of-domain data. Although an in-domain corpus exists for EN–DE in the additional resources, it represents a mix of datasets resulting from a different distribution compared to the training data in Table 1. For this reason, all corpora in the additional resources are considered out-of-domain data.

The perplexity computed on the target side of the development set using all available data is 207. When applying both data selection methods, it significantly decreased to 150, indicating that selecting data in this fashion can be advantageous. The entropy-based method achieved a perplexity of 150, and selecting only the top 15% of the ranked sentences resulted in 3.3 million sentence pairs. The latent domain method obtained a similar perplexity (157) but selected a larger number of sentences. For this reason, the entropy-based technique is also used for EN–CS. In this case, the perplexity is higher than for EN–DE (1900), but using the top 5% of the ranked data (2.5 million sentences) allowed us to significantly reduce it to 1300. These high perplexity values stem from the fact that the external resources for EN–CS do not contain any IT data.

3.2 MT Systems

Different systems were built for each language pair using the selected and the in-domain data for EN–DE and EN–CS and the in-domain data for the other language pairs.

- EN–DE: Two different MT systems were created: a PBMT and a NMT system. The PBMT system was trained on all of the selected parallel training data. The phrase table was adapted to the in-domain data using the approach proposed in (Niehues and Waibel, 2012). To deal with complex reordering in the German language, this system uses a pre-reordering technique (Herrmann et al., 2013) in combination with lexical reordering. In addition, it takes advantage of two word-based n-gram language models and three additional non-word language models, namely, two automatic word class-based (Och, 1999) language models using 100 and 1,000 word classes, and a POS-based language model using fine-grained POS tags (Schmid and Laws, 2008). For the NMT system, we trained the Nematus toolkit (Sennrich et al., 2017) which is an implementation of the attentional encoder-decoder architecture (Bahdanau et al., 2014). To handle large vocabulary, the training data was previously segmented using the byte-pair encoding compression algorithm (Sennrich et al., 2016), resulting in a vocabulary of 40,000 sub-word units for both languages. We used mini-batches of 100, word embeddings of 500 dimensions, and gated recurrent unit layers of 1,024 units. The maximum sentence length was set to 50. The models were trained using Adam and by reshuffling the training set at each epoch. The NMT system was trained on the selected data and then fine-tuned on the in-domain data.

- EN–CS: The PBMT system was trained using Moses (Koehn et al., 2007) combined with TectoMT (Žabokrtskỳ et al., 2008). This was done by adding the source development and test sentences and their translations obtained by TectoMT as additional (synthetic) parallel

EN–DE		EN–CS	EN–LV		DE–EN
PBMT	NMT	PBMT	PBMT	NMT	PBMT
35.9	45.8	38.7	46.5	38.4	53.4

Table 4: BLEU score of the PBMT and NMT systems on different language pairs.

data to the Moses system previously trained on the selected data. This new corpus and the in-domain data were used to train separated phrase tables. At test time, we ran Moses using all of the phrase tables and we corrected its output using Depfix (Rosa et al., 2012). In addition, we trained a 7-gram LM on surface forms from all monolingual resources. Similar to the EN–DE system, two additional LMs over morphological tags were built to help maintain morphological coherence in the translation output. The system is described in (Tamchyna et al., 2016).

- EN–LV: The PBMT system was trained on Tilde's MT platform (Vasiļjevs et al., 2012). The system is based on the Moses toolkit using the standard components. Nematus with sub-word units was used to train the NMT system with a vocabulary size of 40,000 sub-words. The models were trained with a projection (embedding) layer of 500 dimensions, recurrent units of 1024 dimensions, a batch size of 20 and dropout enabled. All other parameters were set to their default values.

- DE–EN: The PBMT system was trained using the same components and adaptation techniques as those used for the EN–DE model.

The results of the different systems for each of the language pairs are reported in Table 4 according to BLEU (Papineni et al., 2002). The parameters of the models were optimised on the development set and the final results computed on the test set. When comparing the PBMT and NMT performance, we noticed that when using a large collection of training data (*i.e.* EN–DE) the NMT system can significantly outperform the PBMT as shown in several evaluation campaigns. However, when the training data is limited (*i.e.* EN–LV) , the PBMT performs better than the NMT. The language pairs with the lowest out-of-vocabulary rate (EN–LV: 0.2 and DE–EN: 0.5) achieve the best BLEU score values. The DE–EN system obtains better performance compared to EN–LV because it can leverage more in-domain training data.

4 Post-Editing Process

Post-editing was performed using the PET tool (Aziz et al., 2012). This is a simple and freely available, open-source tool that tracks PE using a number of indicators. Figure 1 shows a screenshot of the tool with an English–German PE task. The tool tracks the process of PE, records PE time per sentence, and logs all keystrokes pressed by the annotator. This allows us to reproduce the PE activity, which can be useful for research on topics such as PE process, productivity gains, and automatic PE. The following information was recorded during PE:

- Editing time: time spent translating or editing a unit.
- Keystrokes: number of keys pressed during the PE according to type of keys (deletion, alpha-numeric, etc.).
- HTER: edit distance between the draft translation and its post-edited version.
- Evaluation: quality assessment based on a pre-defined set. We ask a question about the usefulness of the draft translation for PE (top left corner in Figure 1).

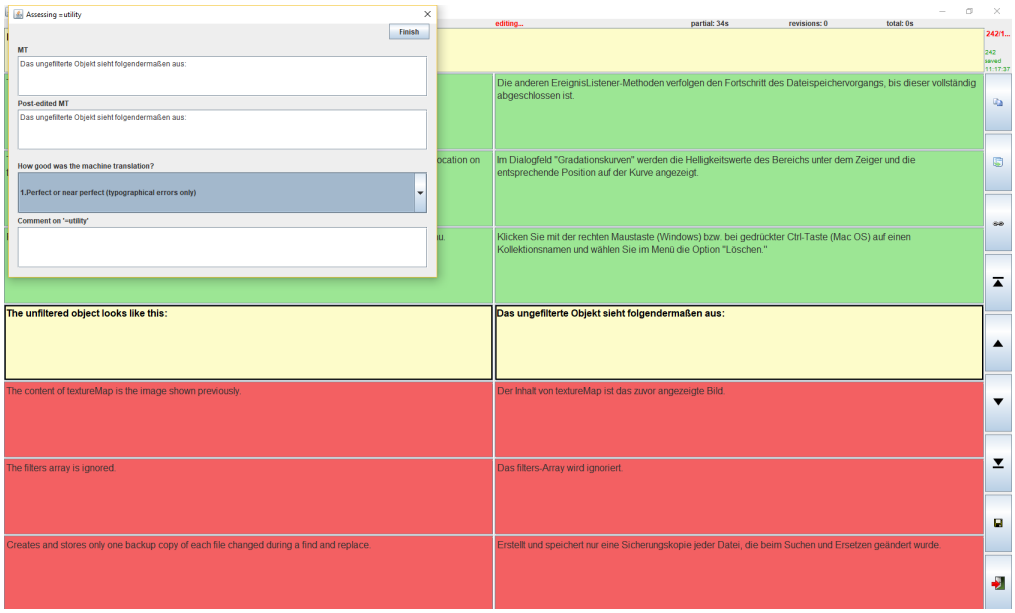


Figure 1: Example of project in the PET tool.

Time, one of the most important indicators collected by the tool, is computed from the moment the target box of the unit is clicked to the moment the task is completed (either the job is closed or the navigation button "next" is pressed). The tool allows for multiple revisions, where the annotator can go back to the same sentence and edit it again. For the statistics reported here, we take the aggregation of PE time and keystrokes, and compute the edit distance between the last version and draft MT output. The outcome of a job is also stored in an XML file.

A set of PE and annotation guidelines created by the QTLaunchpad project were adapted for the PE of our data. To ensure that the quality of the post-edits was consistent and reflected the requirements of the research to be performed on the resulting data, agreement was reached on the level of editing to be done on the data. Based on the previous experience of the language partners involved, the following general rules were defined:[5]

- Use as much of the raw MT output as possible.
- Aim for grammatically and syntactically correct translations.
- Ensure that no information has been accidentally added or omitted.
- Edit any offensive, inappropriate or culturally unacceptable content.
- Ensure proper and appropriate spelling.
- Do not restructure or change word order solely to improve the flow of the text unless dictated by grammar or domain standards.

Additionally, the following domain-specific rules for software localisation were used:

- Ensure that domain-specific terminology is correctly translated.
- Ensure that standard domain and language-specific style issues are followed.
- If formatting is used, ensure that it is correct.

[5]For details: `qt21-wiki.dfki.de/index.php?title=Post-editing_guidelines`

	# sentences		# words SRC	# words MT		# words PE	
Lang.	PBMT	NMT	-	PBMT	NMT	PBMT	NMT
DE–EN	45,000	–	14.66	15.54	–	15.58	–
EN–DE	30,000	15,000[6]	14.47	14.61	14.77	14.61	14.56
EN–LV	20,738	20,738	15.91	13.50	13.42	13.52	13.42
EN–CS	45,000	–	15.04	13.16	–	13.23	–

Table 5: General statistics of the post-edited data: Total number of sentences, average number of words in source, translation and post-edited sentences.

Avg. utility score		
Lang.	PBMT	NMT
DE–EN	1.62	–
EN–DE	1.98	1.40
EN–LV	1.64	1.84
EN–CS	2.17	–

Table 6: PE utility scores: the lower the score, the more useful the MT output.

	Avg. TER MT-PE		Avg. TER REF-PE	
Lang.	PBMT	NMT	PBMT	NMT
DE–EN	0.17	–	0.36	
EN–DE	0.25	0.08	0.40	0.37
EN–LV	0.15	0.23	0.29	0.34
EN–CS	0.32	–	0.34	–

Table 7: Average edit distance between PE and original MT (HTER), and between PE and independent reference. The higher the distance, the more edits performed.

The guidelines were made available to all language teams and pre-editing meetings were held to avoid communication issues. Consistency in the application of these rules was critical, which is why professional translators were employed and thorough consultations were performed prior to PE.

Professional translators performed PE on every language pair. Six translators were involved in the PE for EN–DE, 4 for DE–EN, 8 for EN–LV, and 5 for EN–CS. For the evaluation score, the following options were given to the translator after the post-editing of each sentence:

- **1.** Perfect or near perfect (typographical errors only).
- **2.** Very good, could be post-edited quickly.
- **3.** Poor, required significant post-editing.
- **4.** Very poor, required retranslation.

Tables 5–8 summarise the outcome of the PE process. Much more detailed information is available in the XML output files. Table 5 provides general statistics on numbers of sentences and words per language pair and MT system type. The average perceived PE effort scores are given in Table 6. Table 7 measures the edit distance between MT and PE, and between PE and the original reference (REF). Finally, 8 shows average PE time and keystrokes. As expected, PE time varies considerably for different sentences, even if outliers are removed. Therefore, Table 8 also shows standard deviations.

5 Error Annotation Process using MQM

Our error annotation process follows a 2-step workflow. After PE, the quality of each sentence is evaluated on a scale from 1–4 as explained in the previous section. A subset of sentences scored as 2 (very good) are then selected for the error annotation phase, during which all issues resolved during the PE phase are classified. The errors are annotated using the Multidimensional Quality Metrics (MQM) error annotation framework (Lommel et al., 2014), which is popular in industry and research, and actively supported by XTM, Trados Studio, and other commercial tools. We

	Avg. PE time		Avg. PE time w/o outliers		Avg. keystrokes	
Lang.	PBMT	NMT	PBMT	NMT	PBMT	NMT
DE–EN	42±80	–	36±45	–	24.71	–
EN–DE	51±78	46±602	46±39	32±36	15.55	13.89
EN–LV	27±77	43±406	23±28	36±39	18.91	26.08
EN–CS	44±43	–	42±35	–	45.78	–

Table 8: Post-editing time and keystrokes: average number of seconds per word, with and without outliers (plus standard deviation) and average number of keys pressed during post-editing of a sentence. Outliers are sentences that took more than four minutes to be edited.

used the open-source tool translate5[7] (see Figure 2), a database-driven tool with a GUI. Source texts, translations, post-edits, and error annotations are organised in a relational database. The tool, originally implemented as a proofreading and PE environment for the translation industry, has been recently extended to support MQM annotation.

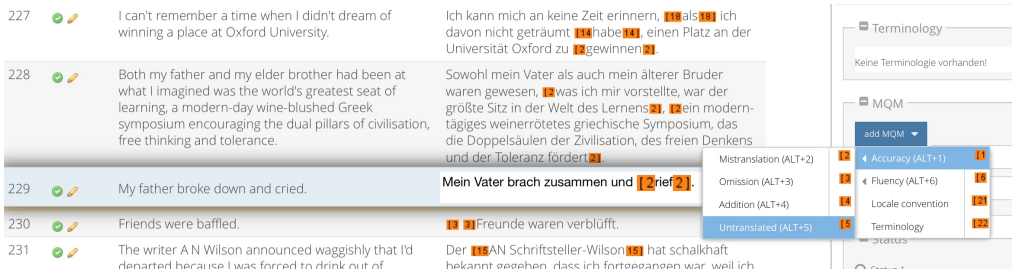


Figure 2: MQM error annotation in translate5 (excerpt of screenshot).

An error represents any issue that has been corrected during the PE step in the translated sentence. In the annotation step, a relevant error classification must be provided for all corrections made during PE according to a given list of errors. Error annotation is performed by experienced professional translators supported by detailed annotation guidelines.

The list of errors is divided into the main issue categories *accuracy*, *fluency* and *terminology*, which fold into a selection of more detailed categories from the MQM hierarchy. Figure 3 shows part of a decision tree that annotators used to select the most appropriate issue. The actual error categories used in the annotation are shown in Table 9.

Annotators are instructed to use the subcategories whenever possible and to resort to the more general category level only in case of doubt, for example, if the German term *Zoomfaktor* is incorrectly translated as *zoom shot factor*, and the annotator is unsure whether this represents a *mistranslation* or an *addition*. In this case, the error can be classified as an Accuracy error since it is unclear whether content has been added or a term mistranslated.

The annotation process has been completed for all languages and MT system types, resulting in 1,800 unique sentences per language pair and MT system type, with an additional 200 sentences doubly annotated for agreement analysis. The breakdown of error annotations for all 2,000 sentences per language pair and MT system type is shown in Table 9.

Table 10 shows an initial analysis on the agreement between pairs of annotators. Agreement was computed using Cohen's kappa (Cohen, 1960) at the word level in two ways: firstly, for each word we count an agreement whenever both annotators agree that it is incorrect (or correct), with agreement by chance = $1/2$; second, for each word we count an agreement whenever

[7] http://translate5.net

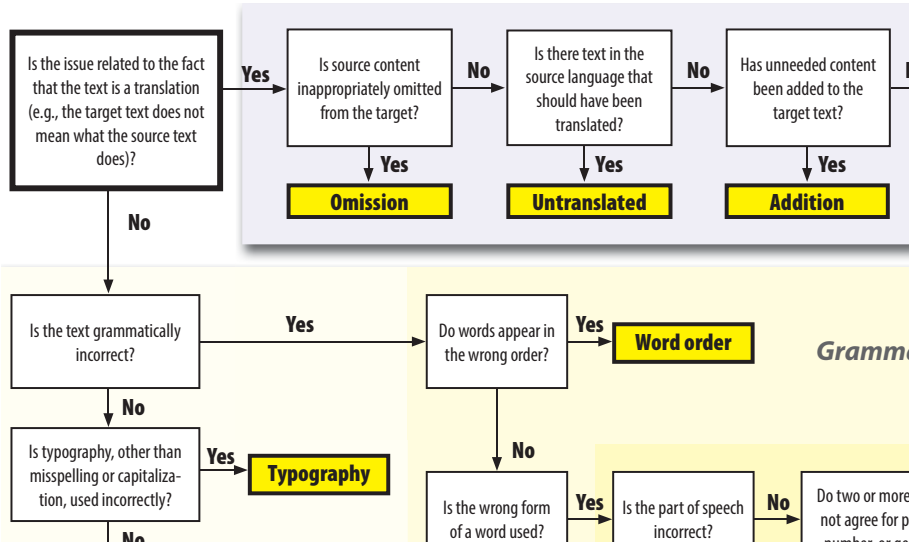


Figure 3: Decision tree guiding error annotation (excerpt).

both annotators agree on the exact error type assigned to the word (or agree on the word being correct), considering all the 20 categories shown in Table 9 as equally likely (i.e. no distinction was made among different levels in the hierarchy), with agreement by chance = $1/21$.

The interpretation of the kappa coefficient is difficult, but it is generally believed that 0.4–0.6 is moderate, while 0.6–0.8 represents substantial agreement, with anything above 0.8 indicating perfect agreement (Landis and Koch, 1977). Considering the subjectivity of the task and the number of error categories and different levels in the hierarchy, we consider the moderate to high agreement found a very positive result towards validating the annotation of the data. In the near future, further quantitative and qualitative analysis will be performed to understand problematic categories and the reasons behind certain disagreements.

6 Examples of Uses of the Dataset

Subsets of the datasets collected have been used in the 2016 and 2017 editions of the WMT shared tasks on Quality Estimation and Automatic Post-editing (Bojar et al., 2016a, 2017).[8] In what follows we summarise some of the outcomes from these tasks.

6.1 Quality Estimation

Quality Estimation (QE) is the task of predicting the quality of the output of an MT system without the use of reference translations (Blatz et al., 2004; Specia et al., 2009). This is approached as a machine learning task, where training data with quality labels is needed. These labels can target different granularity levels: words, phrases, sentences or entire documents.

Early work in the area relied on proxies to quality labels generated using automatic evaluation metrics such as BLEU (Papineni et al., 2002) based on human translations. The task was thus framed as that of predicting an automatic evaluation metric score. This did not prove very successful because of the limitations of the automatic metrics themselves and the lack of a clear interpretation for the predictions (i.e. what does a BLEU score of 0.5 mean?).

Quality labels given by humans have been suggested in (Quirk, 2004) but only started to be

[8] `http://www.statmt.org/wmt16/` and `http://www.statmt.org/wmt17/`.

	DE–EN	EN-DE		EN–LV		EN–CS
Error type	PBMT	PBMT	NMT	PBMT	NMT	PBMT
Accuracy	3	0	0	39	50	0
Addition	539	332	167	277	268	385
Mistranslation	437	967	852	274	677	786
Omission	576	690	355	395	560	588
Untranslated	278	102	24	79	62	301
Fluency	3	0	0	233	210	234
Grammar	0	0	0	11	2	103
Function words	1	2	1	0	0	0
Extraneous	302	525	245	49	49	228
Incorrect	139	804	449	56	55	454
Missing	362	779	231	66	32	348
Word form	0	94	267	280	261	1401
Part of speech	20	128	132	38	35	147
Agreement	18	506	97	419	357	48
Tense/aspect/mood	63	184	51	60	46	397
Word order	218	868	309	336	152	1148
Spelling	118	126	132	324	387	638
Typography	282	553	249	823	387	1085
Unintelligible	0	33	0	10	14	30
Terminology	27	82	139	34	31	0
All categories	3386	6775	3700	3803	3635	8321

Table 9: MQM error categories and breakdown of annotations completed to data.

	DE–EN	EN-DE		EN–LV		EN–CS
	PBMT	PBMT	NMT	PBMT	NMT	PBMT
# annotated words (A1/A2)	516/643	974/920	338/288	669/682	303/310	324/370
Kappa on annotated words	0.61	0.70	0.82	0.69	0.67	0.62
Kappa on error type	0.51	0.48	0.69	0.53	0.51	0.51

Table 10: Number of annotated words per language pair for each annotator (A1 and A2) and the Cohen's kappa measuring inter-annotator agreement for MQM error annotations.

used more recently (Specia et al., 2009). In particular, the use of objective labels derived from extrinsic uses of MT output, such as PE, have become popular (Specia, 2011). Labels of this type include normalised PE distance (HTER - Human Targeted Translation Error Rate (Snover et al., 2006)). These can be acquired as a by-product of PE in a translation workflow, are less subjective and less subject to biases such as the annotators' perception of MT.

The datasets described in this paper open many new avenues for research in QE. The main benefits with respect to previously collected labels include its scale, domain specificity and the availability of multiple types of (reliable) human annotation.

In the WMT16 QE shared task, a subset of the English-German IT domain post-edited data containing 15,000 sentences was used for the sentence, word and phrase-level tasks. The quality labels were automatically derived from the PE of the MT output, e.g. for sentence level, HTER scores were used. Bojar et al. (2016a) claim that, when compared to previous year – approx. 14,000 crowdsourced post-edited sentences – the results of the 2016 task were more conclusive. They attribute this to the higher quality of the new dataset and observe that:

Task	Baseline ↑	Best system ↑
2016 Training set		
Word-level QE	0.32	0.55
Phrase level QE	0.40	0.50
Sentence-level QE	0.35	0.53
2017 Training set		
Word-level QE	0.36	0.58
Phrase level QE	0.33	0.60
Sentence-level QE	0.39	0.71

Table 11: QE shared task results on the 2016 test set: baseline and winning systems in 2016 and 2017 (larger training set) for sentence (Pearson), word and phrase (F_1-mult = multiplication of F_1 for the GOOD and BAD classes) levels.

- for sentence level, the best Pearson correlation between the system prediction and true HTER in 2015 was 0.39 (against 0.14 of the baseline system). In 2016, the winning submission reached 0.52 Pearson correlation (against 0.35 of the same baseline system). One can speculate that the task was made somewhat "easier" by using high quality data, but the delta in the Pearson correlation between the baseline and winning submission is still substantial.
- for word level, 2016 systems performed much better: 0.56 against 0.43 F_1-BAD. The baseline systems are not comparable.

In order to further push progress in the QE field, the 2017 QE task was provided with an extended version of the 2016 dataset in addition to data from a different domain and a different language pair. For English-German, the 2016 dataset was extended to include a total of 28,000 sentence pairs. For German-English, 28,000 sentence pairs in the life sciences domain were made available for the task.

The two datasets are significantly larger than any dataset used before in QE shared tasks. The same data was used for the three subtasks: sentence, word and phrase levels. The results of this year's task (Bojar et al., 2017) show major improvements for all tasks over the 2016 results. In addition to general advances in the field, these can in part be attributed to the larger dataset provided. For the 2016 test set, also used in 2017 for comparison, Table 11 shows the results using the official metrics for the best system and the baseline system using the 2016 vs the 2017 training sets.

This data has proven useful for subsequent work in the field: for instance, (Forcada et al., 2017) focuses on the prediction of PE time at sentence level on the 2016 dataset, while (Martins et al., 2017) proposes an novel word-level QE approach using automatic PE techniques.

6.2 Automatic Post-Editing

Automatic Post-editing (APE) systems are usually trained on (*source, MT, human_post-edit*) triplets from which the appropriate corrections of systematic errors should be learned and possibly generalised. This supervised learning problem is addressed as a "monolingual translation" task in which rough MT output in a given target language has to be translated into a fluent and adequate translation of the original source text. BLEU and TER computed against reference human post-edits are the standard evaluation metrics for the task, and their respective improvements and reductions are usually compared against the baseline scores obtained by the original MT output that has been left untouched (*i.e.* rough, non post-edited translations).

Early APE systems (Allen and Hogan, 2000; Simard et al., 2007) were developed under the PBMT paradigm, that is, by learning from "parallel" data, either (*MT, human_post-edit*)

pairs or triplets including information from the source text (Béchara et al., 2011; Chatterjee et al., 2015). Recent solutions achieved larger and more significant improvements by exploiting neural methods (Junczys-Dowmunt and Grundkiewicz, 2016; Pal et al., 2016, 2017), which approach the task as a sequence to sequence learning problem.

Both paradigms suffer from drawbacks that have, to date, represented the main obstacles towards a wider adoption of APE technology. According to Bojar et al. (2015), one of the major problems lies in data sparsity, which limits the ability to exploit training data in order to learn correction patterns that can also be applied to test instances. Several factors contribute to raising this data sparsity issue, namely: *i)* the size of the data (although human post-edits are a by-product of industrial translation workflows, few corpora are available for research, *ii)* the domain of the data (general domains – like *news* – are definitely less repetitive than narrow ones – like *information technology*), and *iii)* the origin of the post-edits (professional post-editors are definitely more reliable and coherent than non-expert ones).

The datasets described in this paper aim to mitigate the problems related to data sparsity for reasons that are similar to those discussed in the previous section on QE. Indeed, their size, domain specificity and professional PE quality may explain the renewed interest and the impressive progress of APE research in the past few years. The following figures drawn from the WMT experience support our claims:

- Number of tasks and submitted runs. At WMT 2016, only one English-German translation task in the IT domain was organised, while 2017 saw two tasks: English-German (IT) and German-English (life sciences). The new corpora (more repetitive than *news* data edited by non-experts in 2015) motivated more teams to participate: from 7 submissions in 2016 to 20 in 2017.
- Improvements over the baseline. The switch to new data coincided with significant performance gains that prove the viability of APE in domain-specific settings. While in 2015 none of the participants was able to beat the baseline, the best English-German submissions in 2016 and 2017 improved over the baseline by up to 5.5 and 7.6 BLEU points.
- Improvements over the PBMT approach. While in 2015 all systems followed this paradigm, falling in the same range of performance, the combination of advancements in neural research and the provision of more suitable data resulted in impressive performance gains in the next two evaluation rounds. The same PBMT system used for comparison in all the evaluation rounds was significantly outperformed by most of the participants in 2016 (up to 3.2 BLEU points) and in 2017 (up to 7.1 BLEU points).

7 Conclusions

In this paper we introduced a large and unique set of data points derived from *industry data* that have been post-edited and annotated by *professional translators*. This allows for specific features and **novel combinations of features** to be used for a variety of research and user-oriented purposes, including establishing the actual PE effort by translators based on time and keystrokes and comparing these results to the *perceived* level of quality of the post-edited sentence, establishing correlations between certain characteristics such as sentence length and post-editing time, or post-editing time and human or automatic quality evaluation metrics. The datasets also measure post-editing productivity and can be used to detect error patterns in the MT output. In addition, the creation of MQM-annotated subsets of these post-edits for typical industry domains provide information about error patterns and support feature-oriented quality estimation and evaluation, among many other novel avenues for research. This dataset is freely available and can be downloaded from the project website: `http://www.qt21.eu/`.

Acknowledgements

This work was supported by the QT21 project (H2020 No. 645452).

References

Allen, J. and Hogan, C. (2000). Toward the Development of a Post Editing Module for Raw Machine Translation Output: A Controlled Language Perspective. In *Third International Controlled Language Applications Workshop (CLAW-00)*, pages 62–71.

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on empirical methods in natural language processing*, pages 355–362. Association for Computational Linguistics.

Aziz, W., de Sousa, S. C. M., and Specia, L. (2012). Pet: a tool for post-editing and assessing machine translation. In *Eighth International Conference on Language Resources and Evaluation*, LREC, pages 3982–3987, Instanbul, Turkey.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Béchara, H., Ma, Y., and van Genabith, J. (2011). Statistical Post-Editing for a Statistical MT System. In *Proceedings of the 13th Machine Translation Summit*, pages 308–315, Xiamen, China.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., and Ueffing, N. (2004). Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, number 315 in COLING '04, Geneva, Switzerland.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., and Tamchyna, A. (2014). Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., and Zampieri, M. (2016a). Findings of the 2016 conference on machine translation. In *First Conference on Machine Translation, Volume 2: Shared Task Papers*, WMT, pages 131–198, Berlin, Germany.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L., and Turchi, M. (2017). Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., and Turchi, M. (2015). Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal.

Bojar, O., Dušek, O., Kocmi, T., Libovický, J., Novák, M., Popel, M., Sudarikov, R., and Variš, D. (2016b). CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Text, Speech and Dialogue: 19th International Conference, TSD 2016, Brno, Czech Republic, September 12-16, 2016, Proceedings*. Springer Verlag. In press.

Chatterjee, R., Weller, M., Negri, M., and Turchi, M. (2015). Exploring the Planet of the APEs: a Comparative Study of State-of-the-art Methods for MT Automatic Post-Editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156–161, Beijing, China.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Cuong, H. and Simaan, K. (2014). Latent domain translation models in mix-of-domains haystack. In *Proceedings of COLING*, pages 1928–1939.

Eetemadi, S., Lewis, W., Toutanova, K., and Radha, H. (2015). Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29(3-4):189–223.

Forcada, M. L., Sánchez-Martínez, F., Esplà-Gomis, M., and Specia, L. (2017). Towards optimizing mt for post-editing effort: Can BLEU still be useful? *Prague Bull. Math. Linguistics*, 108:183–195.

Gao, J. and Zhang, M. (2002). Improving language model size reduction using better pruning criteria. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 176–182. Association for Computational Linguistics.

Herrmann, T., Niehues, J., and Waibel, A. (2013). Combining Word Reordering Methods on different Linguistic Abstraction Levels for Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, Altanta, Georgia, USA.

Junczys-Dowmunt, M. and Grundkiewicz, R. (2016). Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation*, pages 751–758, Berlin, Germany.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Lommel, A. R., Burchardt, A., and Uszkoreit, H. (2014). Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. *Tradumàtica: tecnologies de la traducció*, 0(12):455–463.

Martins, A. F. T., Junczys-Dowmunt, M., Kepler, F., Astudillo, R., and Hokamp, C. (2017). Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics (to appear)*.

Moore, B. (2002). Fast and accurate sentence alignment of bilingual corpora. Springer-Verlag.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224. Association for Computational Linguistics.

Niehues, J. and Waibel, A. (2012). Detailed Analysis of Different Strategies for Phrase Table Adaptation in SMT. In *Proceedings of the 10th Conference of the Association for Machine Translation in the Americas*, San Diego, CA, USA.

Och, F. J. (1999). An Efficient Method for Determining Bilingual Word Classes. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, Bergen, Norway.

Pal, S., Naskar, S. K., Vela, M., Liu, Q., and van Genabith, J. (2017). Neural automatic post-editing using prior alignment and reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 349–355, Valencia, Spain.

Pal, S., Naskar, S. K., Vela, M., and van Genabith, J. (2016). A neural network based approach to automatic post-editing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 281–286, Berlin, Germany.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Quirk, C. (2004). Training a sentence-level machine translation confidence measure. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

Rosa, R., Mareček, D., and Dušek, O. (2012). Depfix: A system for automatic correction of czech mt outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pages 362–368, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rousseau, A. (2013). XenC: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, 100:73–82.

Schmid, H. and Laws, F. (2008). Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-Grained POS Tagging. In *International Conference on Computational Linguistics (COLING 2008)*, Manchester, Great Britain.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli Barone, A. V., Mokry, J., and Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

Simard, M., Goutte, C., and Isabelle, P. (2007). Statistical Phrase-Based Post-Editing. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT)*, pages 508–515, Rochester, New York.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA.

Specia, L. (2011). Exploiting objective annotations for measuring translation post-editing effort. In *15th Conference of the European Association for Machine Translation*, EAMT, pages 73–80, Leuven, Belgium.

Specia, L., Turchi, M., Cancedda, N., Dymetman, M., and Cristianini, N. (2009). Estimating the Sentence-Level Quality of Machine Translation Systems. In *13th Annual Conference of the European Association for Machine Translation*, EAMT, pages 28–37, Barcelona, Spain.

Tamchyna, A., Sudarikov, R., Bojar, O., and Fraser, A. (2016). Cuni-lmu submissions in wmt2016: Chimera constrained and beaten. In *Proceedings of the First Conference on Machine Translation*, pages 385–390, Berlin, Germany. Association for Computational Linguistics.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In *LREC*, volume 2012, pages 2214–2218.

Vasiļjevs, A., Skadiņš, R., and Tiedemann, J. (2012). Letsmt!: a cloud-based platform for do-it-yourself machine translation. In *Proceedings of the ACL 2012 System Demonstrations*, pages 43–48. Association for Computational Linguistics.

Wisniewski, G., Singh, A. K., Segal, N., and Yvon, F. (2013). Design and analysis of a large corpus of post-edited translations: Quality estimation, failure analysis and the variability of post-edition. In *Proceedings of the MT Summit*, pages 117–124.

Žabokrtskỳ, Z., Ptáček, J., and Pajas, P. (2008). Tectomt: Highly modular mt system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170. Association for Computational Linguistics.

Zhechev, V. (2012). Machine translation infrastructure and post-editing performance at autodesk. In *AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP 2012)*, pages 87–96.

The Microsoft Speech Language Translation (MSLT) Corpus for Chinese and Japanese: Conversational Test data for Machine Translation and Speech Recognition

Christian Federmann chrife@microsoft.com
William D. Lewis wilewis@microsoft.com
Microsoft Translator
Microsoft AI+Research, Redmond, WA, USA

Abstract

Recent years have seen unprecedented growth in the use of MT across industries and domains. Partly this is due to the ready availability of open source MT tools such as Moses or online or customizable services. It is also due to fundamental shifts in the technology, specifically the move to deep learning, which has dramatically improved the quality of MT engines, including those used by online services. Likewise, improvements in Speech Recognition (SR) technology, also driven by the move to deep learning, are showing significant improvements in quality driven by deep learning alone. The improvements of both of these technologies, MT and SR, increase the potential viability for speech translation, since the error cascade caused by daisy-chaining these technologies drops as the quality bar raises. MT is a crucial component in speech translation systems, yet developing conversational MT systems essential to speech translation is not a focus for many working in the Machine Translation discipline. Particularly problematic for many languages is the absence of test and dev data, not any less true for the Chinese and Japanese languages, where forays into conversational MT in and out of these languages are limited by the lack of publicly available conversational test data. In this paper, we seek to address this problem, by providing MT test and dev data that has been built from actual bilingual conversations between English and Japanese and Chinese, test data that can be useful to drive further research in this space for these two languages. Our plan is to make the data described in this paper available to the public by MT Summit.

1 Introduction

The commoditization of MT, as evidenced by increased use of MT across industries and domains, results most significantly by the availability of open source MT tools such as Moses [1] and online tools for training and building customized systems (such as those offered by Microsoft, SDL, IBM, etc.). But it is also due to fundamental

shifts in the technology, specifically the move to deep learning, which has dramatically improved the quality of MT engines [2, 3, 4], including those used by online services (*e.g.*, Google, Microsoft, Baidu, etc.). Likewise, improvements in speech recognition technology, also driven by the move to deep learning, are showing 25-50% improvements in quality driven by deep learning alone; for instance, [5] showed a 32% reduction in Word Error Rate when switching from Gaussian Mixture Models (GMMs) to Deep Neural Networks (DNNs), *with no change in training data*. The improvements of both of these technologies increase the potential viability for speech translation, since the error cascade caused by daisy-chaining these technologies drops as the quality bar in each component technology increases.

MT is a crucial component in speech translation systems, yet developing conversational MT systems essential to speech translation is not a focus for many working in the Machine Translation discipline. With the increase of conversation-like sources that needed to be translated, however, *e.g.*, social media, and an increase in the availability of speech recognition systems across multiple languages, translating less formal content is becoming far more commonplace and in-demand. MT systems that are trained on more "general" content, say, Web page content or parallel PDF documents, do not do well on content of a radically different style [6, 7].

Also problematic for developing conversational MT systems is an adequate way to evaluate them. Our focus in this paper is on test data for Chinese and Japanese, in and out of English. We describe here test and dev data that has been built from actual bilingual conversations between English to and from Japanese and Chinese. This test data consists not only of the audio (not relevant to MT *per se*, but certainly to speech translation), but also "raw" un-edited transcripts of the audio, cleaned-up caption-like transcripts, and translations to/from English and Japanese and Chinese. For each language there are two test sets: one from English, and one to English. This provides data that is native in the source language, eliminating problems with direction-bias in evaluation. Also, because the data is conversational, it is fully appropriate to tune systems to, and evaluate systems on, conversational-style content.

It should be noted that the bilingual English↔Japanese and English↔Chinese conversations were unscripted; participants were given some guidance with respect to topic, but otherwise, were allowed to have unrestricted conversations with one another. In many ways this is similar to the instructions provided to participants in the construction of the monolingual Switchboard corpus [8]. Because the conversations were unscripted, the data is rife with content typical of conversations: filler pause-words (um, uh), discourse markers (you know, I mean), restarts (I'm...I've), stutters (I-I-I), colloquial forms (gonna, kinda), and a host of disfluencies and artifacts common to colloquial speech. See Figure 1 for a few examples of Chinese and Japanese disfluencies. This provides a means to test MT systems designed for less formal, more conversation-like content, not only limited to output from speech recognition, but also content common in social media. Although our test data does not translate disfluencies *a la* [9], which was by design, it does preserve the informal character of the input content in the translation.

Language	Type	Example	Approximate Meaning
Chinese	Pause Word	呃	N/A (like "uh")
Chinese	Discourse Marker	那个	"that one", "which one"
Japanese	Pause Word	は	Topic marker (repeated)
Japanese	Discourse Marker	あのう	"that"

Figure 1: Example disfluencies in Chinese and Japanese.

2 Data Collection

The focus of our work here was to create realistic test data for evaluating conversational MT systems. Thus, we wanted the test data that reflected actual *bilingual* conversations between fluent speakers of English, Chinese and Japanese. Monolingual corpora of this type exist, *e.g.*, LDC2004T19 and LDC2005T19 [10, 11] for English, and the CALLHOME corpora for English and Chinese and other languages [12, 13, 14, 15] (but notably, *not* Japanese). The focus of these corpora, however, are explicitly on Speech Recognition and not Machine Translation, since no translations of the content are publicly available. For those interested in conversational Machine Translation or Speech Translation, these corpora are of little use, unless one wishes to expend significant resources in translation. Further, one has no options from Japanese, since these corpora do not cover Japanese.[1]

The "realistic" test data requirement also meant, crucially, that we did not want our test data to be constrained by the current state of the art in speech recognition and machine translation technology, nor constrained by domain. This requirement meant that we avoided using existing speech recognition and machine translation systems in data collection.

2.1 Recording Guidelines

As noted, we opted not to use existing speech recognition or machine translation technology in our recordings. This was motivated, in part, by experiments we conducted on English, German and French [17]. In these experiments, we noted that users behavior changed dramatically when having conversations mediated by a speech translation system: speech rate dropped dramatically from monolingual conversations, vocabulary was more constrained, conversations were punctuated by a significant number of restarts and rephrases, and users would often ask questions solely for the purpose of clarification (*e.g.*, when ASR or MT failed), affects that would not be present in fluent conversations. *Our interest is in constructing* ***realistic*** *bilingual conversations*, notably not constrained by the current state of the art, specifically sans the artifacts described above. Constraining systems in such a way would set a ceiling to what exists currently, and thus not provide a true gold standard conversational content to evaluate against.

[1]The BTEC corpus [16] does contain quasi-conversational Japanese input, but it is focused on the travel domain, and does not consist of free form conversations and transcripts.

Without the aid of machine translation, we could only recruit bilingual speakers of English, Chinese and Japanese. Bilinguality varies significantly across speakers, but generally, speakers will be dominant in one of the languages (usually their native or mother tongue) and less capable in other language(s). We recruited fluent bilingual speakers, who would naturally understand utterances in either language, but for the source language, only those who were dominant in that language. Thus, Japanese bilingual speakers needed to be native (or dominant) in Japanese, but capable of understanding English. Likewise for Chinese speakers.

For the recordings, no machine translation was used. Users held conversations over communication software installed on their computers, specifically Skype, and we captured the audio from these conversations. The audio data was then segmented into smaller chunks (typically, less than 30 seconds long) and transcribed faithfully, capturing all disfluencies present in the audio signal.[2] For each pair of speakers, we organized recordings adhering to the following paradigm:

- Speakers recorded two sessions, 30 minutes each;
- Speakers switched roles, speaking their native language in one conversation, English in the other;
- Conversations were lightly constrained to predefined topics. Topics were used more to prime conversations than to act as constraints, and included topics such as sports, pets, family, education, food, etc.

We recorded over 100 speakers for each language, with 50+ pairings. Speakers were balanced for gender and age groups. The English side of the recordings for Japanese and Chinese bilingual conversations were discarded as they represented accented speech, and thus less desirable, given our *unrestricted* requirement. In other words, the bilinguals we recruited were dominant in Japanese and Chinese first, English second. For English-only, we collected data from monolingual English conversations between speakers of different English dialects (American, Australian, British and Indian), ensuring speaker and dialect diversity.

2.2 Annotation Guidelines

We asked annotators to transcribe the given audio signal in disfluent, verbatim form. Incomplete utterances and other sounds are transcribed using:

- predefined **tags** such as <SPN/> or <LM/>, and
- free **annotations** such as `[laughter]` or `[door slams]`.

In theory, annotators are free to choose whatever annotations they deemed appropriate for sounds which none of the predefined tags captured. In reality we observed only one such annotation: `[laughter]`.

[2]For capturing the audio, we used a specially modified Skype client that allowed us to record the audio on the local computer, at the same time that they were holding a conversation.

The following list provides details on the predefined tags and their interpretation.

- **SPN: Speech noise:** Any sounds generated during speaking which are not actual words should be transcribed as speech noise. Examples are lip smacks or breathing noises from the primary speaker.
- **EU: End unspoken:** Used when the end of a word was truncated or swallowed by the speaker, possibly due to hesitation. Example: `"hell<EU/> hello"`.
- **NON: Non-speech noise:** Any sounds which are not generated by a speaker should be transcribed as non-speech noise. Examples are external sounds such as cars or music from a TV running in the background.
- **UNIN: Unintelligible:** When the transcriber cannot even make an educated guess at which word has been uttered by the speaker, it should be transcribed as unintelligible. Should be applied to one word at a time. For multiple such words, multiple tags should be used.
- **LM: Language mismatch:** If the word uttered by the speaker is understandable but not in the expected speech language the annotator should use the language mismatch tag. If the foreign word can be identified, it should be embedded into the tag, otherwise an empty tag is sufficient. Examples are `"Hello <LM>monsieur</LM>"` or `"I visited <LM/>"`.[3]
- **AS: Audio spill:** If the audio signal is polluted by feedback or audio bleeding from the second channel or affected by any other technical issues, this should be transcribed as audio spill. Generally, this indicates bad headsets or recording conditions.
- **SU: Start unspoken:** Used when the beginning of a word was truncated or otherwise messed up by the speaker. Example: `"<SU/>an hear you"`.
- **UNSURE: Annotator unsure:** Indicates a word the transcriber is unsure of. Should be applied to one word at a time. For multiple such words, multiple tags should be used.
- **NPS: Non-primary speaker:** Indicates a word or phrase which has been uttered by a secondary speaker. This speaker does not have to be identified. Example: `"watching the water flow. <NPS>yeah.</NPS>"`
- **MP: Mispronounced:** A mispronounced but otherwise intelligible word. Example: `"like, a filet <MP>mignon</MP>"`

Table 2 gives a detailed overview on the observed frequencies of these tags for each of the released MSLT data sets.

3 Corpus Data

The Microsoft Speech Language Translation (MSLT) corpus for Japanese and Chinese will be made available at the following URL:

- `https://aka.ms/mslt-corpus`

[3] In the latter example, taken from real data, the `<LM/>` tag indicates an utterance in a language that the transcriber did not know, and was left untranscribed, *e.g.*, I visited Ceuta.

Language	Data set	Files	Runtime	Average
English	Test	3,304	4h03m58s	4.4s
	Dev	3,052	3h56m37s	4.7s
Japanese	Test	4,160	5h22m23s	4.6s
	Dev	3,179	4h29m07s	5.1s
Chinese	Test	1,285	2h24m36s	6.8s
	Dev	1,256	2h12m28s	6.3s

Table 1: Audio runtime information for our Test and Dev data by source language.

At this site, we provide the audio files (see format description in Section 3.1 below), disfluent and fluent transcripts ("T1" and "T2"), and English translations ("T3"). In addition to the English, Chinese and Japanese corpora, we provide links at this site for the other corpora in the MSLT family of corpora, including the English, German, and French MSLT corpus released last year [17]. We ask that users of the Japanese and Chinese corpora cite this paper when used in their research (and [17] when using the English, French and German corpora). Also, please refer to the license agreement contained in the download packages for details on citation and limits of use.

3.1 Audio Files

The corpus contains uncompressed WAV audio files with the following properties:

- **Encoding:** PCM
- **Sample rate:** 16,000 Hz
- **Channels:** 1, mono
- **Bitrate:** 256 kbit/s

Note that the original audio streams had been encoded using the Siren codec so we had to transcode them to create the uncompressed files for release. Furthermore, the original signal had been subject to transport via Skype's network with variable bandwidth encoding. Audio quality of the released files may be affected by both factors. Files represent a realistic snapshot of speech quality in real life. Table 1 gives more details for the audio portions of the MSLT release.

3.2 Transcription and Translation Files

Transcripts (T1, T2) and translations (T3) are formatted as Unicode (16 bits, little-endian) text files. We defined these three text annotation layers for our speech-to-speech processing:

- **T1: Transcribe:** represents a raw, human transcript which includes all disfluencies, hesitations, restarts, and non-speech sounds. The goal of this annotation

step is to produce a verbatim transcript which is as close to the original audio signal as possible. Audio were provided to annotators segmented at the utterance level. Segmentation was done using an existing ASR engine using a Voice Activity Detection (VAD) algorithm. We observed bias when speakers annotated their own transcripts (repairing, *e.g.*, disfluencies and restarts, or transcribing words based on original intent), so we assigned work to a different set of consultants to prevent this issue. The extra effort regarding transcription resulted in higher transcription fidelity, especially regarding disfluencies, noises and incomplete utterances. Both punctuation and case information are optional in T1 but we found that most annotators already provided this. We assume they added this information to make the subsequent T2: Transform processing easier.

- **T2: Transform:** represents a cleaned up version of the T1 transcript with proper punctuation and case information. Of course, T2 data should not contain any disfluencies or other annotations. T2 output also should be segmented into *semantic units*. While the audio signal has already been segmented using VAD, the resulting utterances typically contain multiple phrases instead of a single sentence. This is partly due to the human speech production process and partly due to deficiencies in our speech segmentation. As machine translation targets individual input sentences, the T1-to-T2 segmentation process is crucial. The idea is to create conversational text which might be printed in a newspaper quote. Segmentation and disfluency removal may introduce phrasal fragments, which are kept as long as they have at least *some* semantic value. Annotators work on the T1 text files only and do not have access to the original audio files. We found that giving the annotators access to the audio signal resulted in longer annotation times, sometimes contradicting the original T1 data, and with less focus on the transformation task.
- **T3: Translate:** represents the translation of the fluent T2 transcript. The goal is to create conversational target text which feels natural to native speakers. Every translation should be usable in a direct quote in a newspaper article. Translations have been created based on unique segments in order to enforce translation consistency. Translators are instructed not to translate any (remaining or perceived) disfluencies but instead asked to flag such T2 instances for repair. The biggest problem for translators was lack of context. Especially for shorter utterances, we observed a lot of ambiguity which made the translation process hard. While we sent out T2 data in order (so that translators could have used contextual cues), any kind of task parallelization will have negatively affected the translation process. Also, our assumption that unique source segments should always have the same target translation might not hold in the case of ambiguous, context-dependent phrases. Our lessons learnt during the original translation process will guide future translation campaigns creating additional references for this data set.

3.3 Corpus Statistics

Table 3 provides an overview on segment, token and type counts for both Test and Dev data for English, Japanese and Chinese. Token length for disfluent T1 transcripts and segmented, fluent T2 transcripts show expected behavior: segment counts increase and the token numbers decrease. Note the significantly higher number of tokens for both English sets. A possible explanation lies in the fact that English conversations were easier as speakers only had to "translate" between different English dialects. Hence,

these conversations were much closer to our monolingual recording scenario than conversations for Japanese or Chinese.

3.4 Some Examples

Figures 2 and 3 give examples containing disfluent, verbatim transcripts (T1), cleaned up and transformed text (T2) and the corresponding translations (T3) into English from both Chinese and Japanese. Note in the Chinese example how T2 transformation breaks the T1 transcript into two segments and also removes disfluencies and annotations. Translations are aligned on the segment level, and only with T2.

Language	Type	Segment	Text
Chinese	T1	1	**呃**，[laughter]，我觉得你，**那个**，在起 <NON/> 锅之前放盐就可以了，<NON/> 就是只要他有咸味就行了嘛
Chinese	T2	1	我觉得你在起锅之前放盐就可以了。
		2	只要他有咸味就行了嘛。
English	T3	1	I think you should put salt in before it's removed from the stove.
		2	As long as it has salty taste, it is good enough.

Figure 2: Examples from the Chinese corpus, with raw transcripts, transformed transcripts, and translations into English. Disfluencies that are removed are highlighted in the source.

Language	Type	Segment	Text
Japanese	T1	1	いや**いやいや**やみついは**はは**正しいこと言ってますはい**あのう**やっぱりね
Japanese	T2	1	いやみついは正しいこと言ってますはいやっぱりね
English	T3	1	No, what you're saying is correct, Matsui

Figure 3: Examples from the Japanese corpus, with raw transcripts, transformed transcripts, and translations into English. Disfluencies that are removed are highlighted in the source.

4 Usage Scenarios

We have previously described the three levels of annotation for the MSLT corpus data. In this section, we will describe how one could use the different annotation layers and explain why all three are needed to evaluate end-to-end quality of a speech translation system.

4.1 Using T1 data: "Bilingual" Speech Recognition

First, our data allows one to measure quality for *bilingual* speech recognition. While the recorded speech data itself is monolingual, our recording setup was bilingual by design. In any given session, both speakers were native speakers of the non-English language, so they could natively understand one other. However, as one of the two had to give answers in English, an additional bilingual element was added to the conversation flow. This affects the conversation. Most notably, we observe a decreased number of words uttered compared to purely monolingual conversations, which makes our data special in this regard and naturally representative of bilingual conversations (rather than monolingual conversations).

Testing speech recognition quality with our data will typically be implemented using word error rate (WER) scoring, comparing an ASR hypothesis against our reference transcription. Depending on the output style of the ASR engine under investigation, the reference text is either T1 or T2 data. Many ASR systems will remove disfluencies and partial recognitions to make resulting transcripts more readable to humans. If testing against such a system, our T2 data should be used as reference for WER scoring. As the segmentation of the T2 reference transcripts will likely not match that of the ASR output (which might not be segmented at all), ASR output should be compared to the "joint" T2 reference, which is the concatenation of all T2 segments into a single line.

Of course, if the ASR system being evaluated does no disfluency processing, then the T1 transcript should be used as the reference for calculating WER.

4.2 Using T2 data: Disfluency Removal

In the construction of the MSLT corpus, we have put in extra effort to annotate and transcribe disfluencies and other non-speech sounds, which are common in conversational speech. Such annotations can be used to evaluate the quality of disfluency removal (DR) models. While it is possible to train machine translation models to learn to translate or otherwise deal with such phenomena—this works pretty well for simple disfluencies, but becomes far more challenging for non-obvious disfluencies or partial utterances and restarts; see [9] for an example of such a system—the data space for these is very sparse. Therefore, we found it more practical to apply a DR component to "clean up" our ASR output before translation, as a normalization step [18].

For evaluation of such DR systems, one would feed the disfluent T1 transcripts into a disflueny removal system and compare the resulting output to the fluent T2 transcripts, which act as the reference. As we have previously discussed, T2 data is both cleaned up (with respect to disfluencies or non-speech noises) and segmented into units that contain at least some semantic value. Doing this will affect the usefulness of the T2 data as references for disfluency removal. If the DR model also performs segmentation, then its output can be directly compared to the T2 references. It has to be noted, however, that even small differences in segmentation will negatively affect the comparison. Hence, it might make more sense to compare the DR output and T2 segments on a non-segmented level. This is similar to the problem of testing "fluentized" ASR output against T2 references, as mentioned above.

4.3 Using T3 data: Conversational Translation

Considering evaluation of machine translation, the main difference to existing test data lies in the conversational nature of the collected data. We are not aware of any data sets which have been produced following the same "bilingual" recording setup. While there are test sets based on conversational speech transcripts, they are typically based on monolingual conversations. Hence, they might not be ideal for testing of bilingual (or even multilingual) conversational MT[4]. The MSLT data is different here as it puts

[4]There are a host of reasons why this might be true: directionality bias (given that one would be translating content from one language to another in one direction but not the other); unnatural

the focus on such bilingual conversation scenarios, albeit emulating a perfect translation component in the form of speakers understanding the non-English language natively. As this approach represents an upper bound on achievable translation quality (subject to the individual language competency of the speakers), the resulting references are perfectly suited for evaluation of conversational translation.

The MSLT corpus data can also be used to evaluate machine translation quality for conversational speech transcripts. To do this, one would use the fluent and segmented T2 transcripts as input segments for an MT system[5]. The resulting output data would then be compared to the corresponding T3 references, using automated metrics such as BLEU or human annotation. As directionality matters for MT evaluation, we provide test sets for translation from English as well as for translation into English. It is important to note that the transcripts for these are from different recording sessions which have been conducted by different speakers. As instructions and recording setup were identical for these, we think that the resulting data represents high quality test data for evaluation of conversational MT.

4.4 End-to-end Speech Translation Systems

Next to testing the performance of components corresponding to the individual annotation layers in the MSLT corpus, its data can also be used for end-to-end testing of speech translation. The setup is straightforward: The system records spontaneous utterances from one or more participants of a conversation. The audio signal is then sent to the speech recognition component which creates disfluent, verbatim transcripts. In a follow-up step, a disfluency removal component removes any disfluencies and separates the input transcript into one or more segments. These segments are fluent and each corresponds to a single "semantic unit", as discussed earlier. In a last step, the fluent segments are translated into the target language. Translation quality is computed based on automated metrics or evaluated using human annotators.

4.5 Multimodal Translation

The MSLT corpus data may also be helpful for multimodal translation. This research area has recently seen increasing interest (as demonstrated by shared tasks at WMT 2016 and 2017 [19]; also as a keynote by Mirella Lapata at ACL 2017) and aims to solve translation problems based on multimodal input. Effectively, our data offers three different input layers (the audio files and the T1/T2 transcripts), all of which are mapped to a single output layer, the T3 translations. It may be possible to build a translation system which uses both the audio signal and the corresponding transcript (likely in a joint, neural network approach) to generate translation output. Quality of such translations can be evaluated using our data set.

---

conversational structures, words, and phrases in the target language; no equivalent set of disfluencies one sees in T1 transcripts; etc.

[5]Again, we point to [9] for an example of where noisy, disfluent transcripts were used as input in a conversational MT system. In such a setting, the MT system itself would be doing much of the disfluency processing, rather than some separate DR module. The upside of such a technique is that the MT system *could* produce relevant disfluencies *in the target language*, given bilingual conversational text data with such disfluencies represented in the data for both languages.

4.6 Evaluation Campaigns using MSLT

MSLT evaluation data for German, French and English was used in the Machine Translation and Speech Recognition tracks at IWSLT 2016[6]. Although participants had access to significant amounts of parallel training data, e.g., from the WMT campaigns[7], they had very limited parallel data for training conversational MT systems. The out-of-the-box MT systems trained on WMT data generally did poorly on MSLT and TED lecture test data. However, adapting the base models using held-out TED data showed significant improvements on both the MSLT and TED test data sets. A notable example are the results from the Karlsruhe Institute of Technology (KIT) submission to IWSLT 2016 [20], where adaptation led to more than 1.5 BLEU score improvements on the MSLT corpus, even though, as the authors noted, the MSLT corpus did not exactly match the TED data used for adaptation (which is lecture-focused, and less conversational).

		English		Japanese		Chinese	
Annotation	**Description**	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**
<SPN/>	Speech noise	200	271	1,445	1,122	722	694
<EU/>	End unspoken	409	388	43	32	20	15
<NON/>	Non-speech noise	192	235	1,446	1,192	987	1,077
<UNIN/>	Unintelligible	306	125	92	110	103	76
<LM/>	Language mismatch	12	0	0	0	44	56
<AS/>	Audio spill	6	0	0	0	11	0
<SU/>	Start unspoken	37	54	10	5	5	1
<UNSURE/>	Annotator unsure	59	81	36	28	24	22
<NPS/>	Non-primary speaker	44	68	3	2	36	27
<MP/>	Mispronounced	3	4	12	20	0	0
[laughter]	Laughter	217	192	31	26	55	43
	Annotations	1,487	1,418	3,471	2,801	2,057	2,067
	Utterances	3,304	3,052	4,160	3,179	1,285	1,256
	Tokens	42,852	41,450	9,169	4,985	3,751	3,804
	Types	36,318	35,308	8,413	4,964	3,510	3,597

Table 2: Annotation information for our Test and Dev data by source language.

5 Conclusion

We presented a corpus of Chinese and Japanese for end-to-end evaluation of speech translation systems and/or component level evaluation. In the latter case, the test data consists of component level data: to test the ASR component, albeit not relevant to MT *per se*, the corpus has audio data and verbatim transcripts; to test disfluency removal and related processing against the raw transcripts—a necessary component if one wishes to process "raw" transcripts coming from, say, an off-the-shelf ASR engine— the corpus

[6]http://workshop2016.iwslt.org
[7]http://www.statmt.org/

Language	Type	Segments	Tokens	Types	Segments	Tokens	Types
English	T1 (EN)	3,304	42,852	36,318	3,052	41,450	35,308
	T2 (EN)	5,175	36,388	31,981	5,313	36,184	31,960
	T3 (JA)	5,175	37,324	33,862	5,313	36,409	32,913
	T3 (ZH)	5,175	39,776	35,614	5,313	40,159	35,824
Japanese	T1 (JA)	4,160	9,169	8,413	3,179	7,333	6,689
	T2 (JA)	5,976	6,221	6,205	4,970	4,985	4,964
	T3 (EN)	5,857	36,853	34,461	4,965	28,105	26,399
Chinese	T1 (ZH)	1,285	3,751	3,510	1,256	3,804	3,597
	T2 (ZH)	2,156	2,208	2,205	2,018	2,097	2,097
	T3 (EN)	2,156	15,665	13,920	2,018	14,284	12,628

Table 3: Segments, tokens and types for our Test/Dev data by source language and annotation type.

has transcripts that have been cleaned up of disfluencies, pause words, discourse markers, restarts, hesitations, laughter, and any other content not relevant to translation; and to test conversational MT, the corpus has translated transcripts into English. We also provide English source with the same characteristics, translated into both Chinese and Japanese. This provides data that facilitates research in conversational MT both into and out of these two languages. It should be noted that the conversations recorded for either direction for any given language pair are not semantically contiguous, that is, they do not consist of recordings of the same conversation sessions. This is due to the fact the English side of Chinese and Japanese conversations was thrown out due to non-English accents, and that all kept English sessions were recorded separately. We feel that the test and dev data that we are providing will be of great use to the community interested in developing conversational MT systems in and out of the Chinese and Japanese languages.

References

[1] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of ACL*, Prague, The Czech Republic, 2007.

[2] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and Robust Neural Network Joint Models for Statistical Machine Translation," in *Proceedings of ACL*, Baltimore, Maryland, 2014, pp. 1370–1380.

[3] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, ukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's Neural Machine Translation System: Bridging

the Gap between Human and Machine Translation," 2016. [Online]. Available: https://arxiv.org/abs/1609.08144

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of ICLR*, 2015.

[5] F. Seide, G. Li, X. Chen, , and D. Yu, "Feature Engineering in Context-Dependent Deep Neural Networks for Conversational Speech Transcription," in *Proceedings of ACRU, IEEE*, 2011.

[6] W. D. Lewis, C. Federmann, and Y. Xin, "Applying Cross-Entropy Difference for Selecting Parallel Training Data from Publicly Available Sources for Conversational Machine Translation," in *Proceedings of the IWSLT 2015*, Danang, Vietnam, December 2015.

[7] P. Wang and H. T. Ng, "A Beam-Search Decoder for Normalization of Social Media Text with Application to Machine Translation," in *Proceedings of NAACL-HLT*, Atlanta, Georgia, June 2013, pp. 471–481.

[8] J. Godfrey and E. Holliman, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, San Francisco, March 1992, pp. 517–520.

[9] G. Kumar, M. Post, D. Povey, and S. Khudanpur, "Some insights from translating conversational telephone speech," in *Proceedings of ICASSP*, Florence, Italy, May 2014. [Online]. Available: http://cs.jhu.edu/~gkumar/papers/kumar2014some.pdf

[10] C. Cieri, D. Graff, O. Kimball, D. Miller, and K. Walker, "Fisher English Training Speech Part 1 Transcripts LDC2004T19," Web Download. Philadelphia: Linguistic Data Consortium, 2004. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2004T19

[11] ——, "Fisher English Training Part 2, transcripts LDC2005T19," Web Download. Philadelphia: Linguistic Data Consortium, 2005. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2005T19

[12] A. Canavan and G. Zipperlen, "CALLHOME Spanish Speech LDC96S35," Web Download. Philadelphia: Linguistic Data Consortium, 1996. [Online]. Available: https://catalog.ldc.upenn.edu/LDC96S35

[13] B. Wheatley, "CALLHOME Spanish Transcripts LDC96T17," Web Download. Philadelphia: Linguistic Data Consortium, 1996. [Online]. Available: https://catalog.ldc.upenn.edu/LDC96T17

[14] A. Canavan, D. Graff, and G. Zipperlen, "CALLHOME American English Speech LDC97S42," Web Download. Philadelphia: Linguistic Data Consortium, 1997. [Online]. Available: https://catalog.ldc.upenn.edu/LDC97S42

[15] P. Kingsbury, S. Strassel, C. McLemore, and R. McIntyre, "CALLHOME American English Transcripts LDC97T14," Web Download. Philadelphia: Linguistic Data Consortium, 1997. [Online]. Available: https://catalog.ldc.upenn.edu/LDC97T14

[16] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversation in the Real World," in *Proceedings of LREC 2002*, Las Palmas, Spain, 2002.

[17] C. Federmann and W. Lewis, "Microsoft Speech Language Translation (MSLT) Corpus: The IWSLT 2016 release for English, French and German," in *Proceedings of the IWSLT 2016*, Seattle, Washington, December 2016.

[18] H. Hassan, L. Schwartz, D. Hakkani-Tur, and G. Tur, "Segmentation and disfluency removal for conversational speech translation," in *In Proceedings of INTERSPEECH 2014*, 2014, pp. 318–322.

[19] L. Specia, S. Frank, K. Sima'an, and D. Elliott, "A shared task on multimodal machine translation and crosslingual image description," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 543–553. [Online]. Available: http://www.aclweb.org/anthology/W/W16/W16-2346

[20] E. Cho, J. Niehues, T.-L. Ha, M. Sperber, M. Mediani, and A. Waibel, "Adaptation and Combination of NMT systems: The KIT Translation Systems for IWSLT 2016," in *Proceedings of the IWSLT 2016*, Seattle, Washington, December 2016.

Paying Attention to Multi-Word Expressions in Neural Machine Translation

Matīss Rikters matiss@lielakeda.lv
Faculty of Computing, University of Latvia

Ondřej Bojar bojar@ufal.mff.cuni.cz
Charles University, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Abstract

Processing of multi-word expressions (MWEs) is a known problem for any natural language processing task. Even neural machine translation (NMT) struggles to overcome it. This paper presents results of experiments on investigating NMT attention allocation to the MWEs and improving automated translation of sentences that contain MWEs in English→Latvian and English→Czech NMT systems. Two improvement strategies were explored—(1) bilingual pairs of automatically extracted MWE candidates were added to the parallel corpus used to train the NMT system, and (2) full sentences containing the automatically extracted MWE candidates were added to the parallel corpus. Both approaches allowed to increase automated evaluation results. The best result—0.99 BLEU point increase—has been reached with the first approach, while with the second approach minimal improvements achieved. We also provide open-source software and tools used for MWE extraction and alignment inspection.

1 Introduction

It is well known that neural machine translation (NMT) has defined the new state of the art in the last few years (Sennrich et al., 2016a; Wu et al., 2016), but the many specific aspects of NMT outputs are not yet explored. One of which is translation of multi-word units or multi-word expressions (MWEs). MWEs are defined by Baldwin and Kim (2010) as "lexical items that: (a) can be decomposed into multiple lexemes; and (b) display lexical, syntactic, semantic, pragmatic and/or statistical idiomaticity". MWEs have been a challenge for statistical machine translation (SMT). Even if standard phrase-based models can copy MWEs verbatim, they suffer in grammaticality. NMT, on the other hand, may struggle in memorizing and reproducing MWEs, because it represents the whole sentence in a high-dimensional vector, which can lose the specific meanings of the MWEs even in the more fine-grained attention model (Bahdanau et al., 2015), because MWEs may not appear frequently enough in the training data.

The goal of this research is to examine how MWEs are treated by NMT systems, compare that with related work in SMT, and find ways to improve MWE translation in NMT. We aimed to compare how NMT pays attention to MWEs during translation, using a test set particularly targeted at handling of MWEs, and if that can be improved by populating the training data for the NMT systems with parallel corpora of MWEs.

The objective was to obtain a comparison of how NMT with regular training data and NMT with synthetic MWE data pays attention to MWEs during the translation process as well

as to improve the final NMT output. To achieve this objective, it needed to be broken down into smaller sub-objectives:

- Train baseline NMT systems,
- Extract parallel MWE corpora from the training data,
- Train the NMT systems with synthetic MWE data, and
- Inspect alignments produced by the NMT.

The structure of this paper is as follows: Section 2 summarizes related work in translating MWEs with SMT and NMT. Section 3 describes the architecture of the baseline system and outlines the process of extracting parallel MWE corpora from the training data. Section 4 provides the experiment setup and results. Finally, conclusions and aims for further directions of work are summarized in Section 5.

2 Related Work

There have been several experiments with incorporating separate processing of MWEs in rule-based (Deksne et al., 2008) and statistical machine translation tasks (Bouamor et al., 2012; Skadiņa, 2016). However, there is little literature about similar integrations in NMT workflows so far.

Skadiņa (2016) performed a series of experiments on extracting MWE candidates and integrating them in SMT. The author experimented with several different methods for both the extraction of MWEs and integration of the extracted MWEs into the MT system. In terms of automatic MT evaluation, this allowed to achieve an increase of 0.5 BLEU points (Papineni et al., 2002) for an English→Latvian SMT system.

Tang et al. (2016) introduce an NMT approach that uses a stored phrase memory in symbolic form. The main difference from traditional NMT is tagging candidate phrases in the representation of the source sentence and forcing the decoder to generate multiple words all at once for the target phrase. Although they do mention MWEs, no identification or extraction of MWEs is performed and the phrases they mainly focus on are dates, names, numbers, locations, and organizations, that are collected from multiple dictionaries. For Chinese→English they report a 3.45 BLEU point increase over baseline NMT.

Cohn et al. (2016) describe an extension of the traditional attentional NMT model with the inclusion of structural biases from word-based alignment models, such as positional bias, Markov conditioning, fertility and agreement over translation directions. They perform experiments translating between English, Romanian, Estonian, Russian and Chinese and analyze the attention matrices of the output translations produced by running experiments using the different biases. Specific experiments targeting MWEs are not performed, but they do point out that using fertility, especially global fertility, can be useful for dealing with multi-word expressions. They report a statistically significant improvement of BLEU scores in almost all involved language pairs.

Chen et al. (2016) use a similar approach as we do. Their "bootstrapping" automatically extracts smaller parts of training segment pairs and adds them to the training data for NMT. The main difference is that they rely on automatic word alignment and punctuation in the sentence to identify matching sub-segments.

3 Data Preparation and Systems Used

To measure changes introduced by adding synthetic MWE data to the training corpora, first, a baseline NMT system was trained for each language pair. The experiments were conducted on English→Czech and English→Latvian translation directions.

Figure 1: Portions of the final training data set for English→Czech

Figure 2: Portions of the final training data set for English→Latvian

3.1 Baseline NMT System

To be able to compare the results with other MT systems, training and development corpora were used from the WMT shared tasks: data from the News Translation Task[1] for English→Latvian and data from the Neural MT Training Task[2] (Bojar et al., 2017) for English→Czech. The English→Czech data consists of about 49 million parallel sentence pairs and the English→Latvian of about 4.5 million. The development corpora consist of 2003 sentences for English→Latvian and 6000 for English→Czech.

Neural Monkey (Helcl and Libovický, 2017), an open-source tool for sequence learning, was used to train the baseline NMT systems. Using the configuration provided by the WMT Neural MT Training Task organizers, the baseline reached 11.29 BLEU points for English→Latvian after having seen 23 million sentences in about 5 days and 13.71 BLEU points for English→Czech after having seen 18 million sentences in about 7 days.

3.2 Extraction of Parallel MWEs

To extract MWEs, the corpora were first tagged with morphological taggers: UDPipe (Ramisch, 2012) for English and Czech, LV Tagger (Paikens et al., 2013) for Latvian. After that, the tagged corpora were processed with the Multi-word Expressions toolkit (Ramisch, 2012), and finally aligned with the MPAligner (Pinnis, 2013), intermittently pre-processing and post-processing with a set of custom tools. To extract MWEs from the corpora with the MWE Toolkit, patterns were required for each of the involved languages. Patterns from Skadiņa (2016) were used for Latvian (210 patterns) and English (57 patterns) languages and patterns from Majchráková et al. (2012) and Pecina (2008) for Czech (23 patterns).

This workflow allowed to extract a parallel corpus of about 400 000 multi-word expressions for English→Czech and about 60 000 for English→Latvian. For an extension of this experiment, all sentences containing these MWEs were also extracted from the training corpus, serving as a separate parallel corpus.

4 Experiments

We experiment with two forms of the presentation of MWEs to the NMT system: (1) we add only the parallel MWEs themselves, each pair forming a new "sentence pair" in the parallel corpus, and (2) we use full sentences containing the MWEs. We denote the approaches "MWE phrases" and "MWE sents." in the following.

4.1 Training Corpus Layout

In both cases, we use the same corpus training corpus layout: we mix the baseline parallel corpus with synthetic data so that MWEs get more exposure to the neural network in training

[1] http://www.statmt.org/wmt17/translation-task.html
[2] http://www.statmt.org/wmt17/nmt-training-task/

Languages	En→Cs		En→Lv	
Dataset	Dev	MWE	Dev	MWE
Baseline	13.71	10.25	11.29	9.32
+MWE phrases	-	-	**11.94**	**10.31**
+MWE sents.	**13.99**	**10.44**	-	-

Table 1: Experiment results.

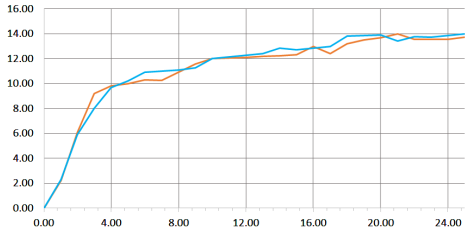


Figure 3: Automatic evaluation progression of En→Cs experiments on validation data. Orange – baseline; blue — baseline with added MWEs.

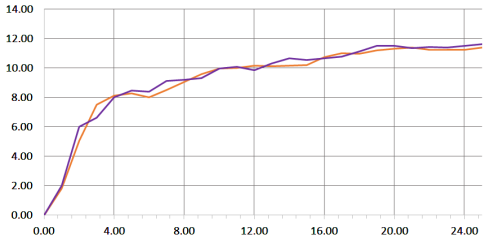


Figure 4: Automatic evaluation progression of En→Lv experiments on validation data. Orange – baseline; purple — baseline with added MWE sentences.

and hopefully allow NMT to learn to translate them better.

Figure 1 and Figure 2 illustrate how the training data was divided into portions. The block 1xMWE corresponds to the full set of extracted MWEs (400K for En→Cs, 60K for En→Lv) and 2xMWE corresponds to two copies of the set (800K for En→Cs, 120K for En→Lv). For En→Lv the full corpus was used. For En→Cs we used only the first 15M sentences to be able to train multiple epochs on the available hardware. The MWEs get repeated five times in both language pairs. By doing this, the En→Cs data set was reduced from 49M to 17M and the En→Lv data set increased to 4.8M parallel sentences for one epoch of training.

While the experiments were running, early stopping of the training was executed and snapshots of the models for evaluation were taken in stages where the models already were starting to converge. For En→Lv this was after the networks had been trained on 25M sentences (i.e. 5.2 epochs of the mixed corpus), for En→Cs 27M sentences (i.e. 1.6 epochs).

Neural Monkey does not shuffle the training corpus between epochs. This is not a problem if the corpus is properly shuffled and the number of epochs is not very large compared to the size of the epochs. We shuffled only the baseline corpus and the interleaved it with (shuffled) sections for MWEs. This worked well when MWEs were provided in full sentences, but not with MWEs presented as expressions. In the latter case, the NMT started to produce only very short output, losing very much of its performance. We, therefore, shuffle the whole composed corpus for the "MWE phrases" runs, effectively discarding the interleaved composition of the training data.

4.2 Results

Table 1 shows the results for both approaches and both language pairs. Due to hardware constraints, we were not able to try out both approaches on both language pairs.

We evaluate all setups with BLEU (Papineni et al., 2002) on the full development set (distinct from the training set), as shown in the column "Dev", and on a subset of 611 (En→Lv) and 112 (En→Cs) sentences containing the identified MWEs (column "MWE").

Sentence	BLEU	Length ratio	Text
Source	-	-	Just like in a city bus or a tram .
Target	100.00	1.00	Stejně jako v městském autobuse či tramvaji .
Baseline	13.54	0.88	jako ve městě autobuse nebo tramvaji .
Improved NMT	41.11	1.00	jen jako v městském autobuse nebo tramvaji .

Figure 5: Differences in translation between baseline and improved NMT system. Improving n-grams are highlighted in green and worsening n-grams — in red.

Sentence	BLEU	Length ratio	Text
Source	-	-	He steps toward the electronic wall map depicting Australia and the surrounding sea areas .
Target	100.00	1.00	Přistoupí k nástěnné elektronické mapě , na níž je znázorněna Austrálie a přilehlé mořské oblasti .
Baseline	10.74	0.75	ukazuje na mapu elektronické stěny zobrazuje Austrálii a okolní mořské oblasti .
Improved NMT	13.87	0.81	vykročil směrem k elektronické mapě , které depicting Austrálie a okolní oblasti .

Figure 6: Differences in translation of a Czech sentence using baseline and improved NMT systems. Improving n-grams are highlighted in green and worsening n-grams — in red.

Figures 3 and 4 illustrate the learning curves in terms of millions of sentences, as evaluated on the full development set.

We see that the difference on the whole development set is not very big for either of the languages, and that it fluctuates as the training progresses.

The improvement is more apparent when evaluated on the dedicated devset of sentences containing multi-word expressions. The improvement for Latvian is even 0.99 BLEU, but arguably, the baseline performance of our system is not very high. Also, more runs should be carried out for a full confidence, but this was unfortunately out of our limits on computing resources.

4.3 Manual Inspection

To find out whether changes in the results are due to the synthetic MWE corpora added, a subset of output sentences from the ones containing MWEs were selected for closer examination. For this task, we used the iBLEU (Madnani, 2011) tool.

In Figure 5, an improvement in the modified NMT translation is visible due to the treatment of the compound nominal "city bus" as a single expression. It seems that the baseline system translates "city" into "městě" and "bus" into "autobuse" individually, resulting in the wrong form of "city" in Czech (a noun used instead of an adjective). On the other hand, the improved NMT translates "city" into "městském" just like the target human translation. Attention alignments will be examined in the following section.

Figure 6 shows an example where the improved NMT scores higher in BLEU points and translates the MWE closer to the human, but loses a part of it in the process. While translating the noun phrase "electronic wall map" the improved system generates a closer match to the human translation "elektronické mapě", it does not translate the word "wall" that was translated into "stěny" by the baseline system. Upon closer inspection, we discovered that this error was caused by the MWE extractor and aligner because the identified English phrase "electronic wall map" was aligned to an identified Czech phrase "elektronické mapě" and the whole phrase

Sentence	BLEU	Length ratio	Text
Source	-	-	It should be noted that this is not the first time that Facebook has been actively involved in determining what network users see in their news feeds .
Target	100.00	1.00	Jāteic , ka šī nav pirmā reize , kad Facebook aktīvi iesaistās , nosakot , ko tīkla lietotāji redz savās jaunumu plūsmās .
Baseline	10.41	1.04	Jāatzīmē , ka šis nav pirmajā reizē , kad Facebook ir aktīvi iesaistīta , nosakot to , ko tīklā izmanto viņu ziņu pārraides .
Improved NMT	27.41	1.13	Ir jāatzīmē , ka šis ir pirmā reize , kad Facebook ir aktīvi iesaistījusies , nosakot to , ko tīkla lietotāji dara viņu ziņu formātā .

Source: It should be noted that this is not the first time that Facebook has been actively involved in determining what network users see in their news feeds.

Baseline: Jāatzīmē, ka šis nav pirmajā reizē, kad Facebook ir aktīvi iesaistīta, nosakot to, ko tīklā izmanto viņu ziņu pārraides.

Improved NMT: Ir jāatzīmē, ka šis ir pirmā reize, kad Facebook ir aktīvi iesaistījusies, nosakot to, ko tīkla lietotāji dara viņu ziņu formātā.

Reference: Jāteic, ka šī nav pirmā reize, kad Facebook aktīvi iesaistās, nosakot, ko tīkla lietotāji redz savās jaunumu plūsmās.

Figure 7: Differences in translation between baseline and improved NMT system. Improving n-grams are highlighted in green and worsening n-grams — in red.

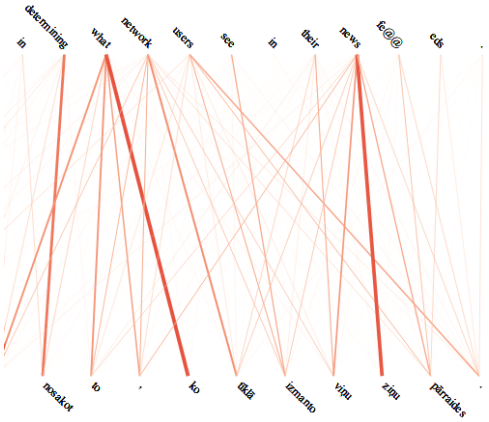


Figure 9: Fragment of soft alignments of the example sentence from the baseline NMT system.

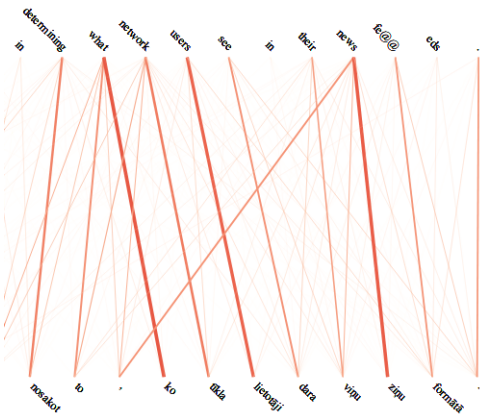


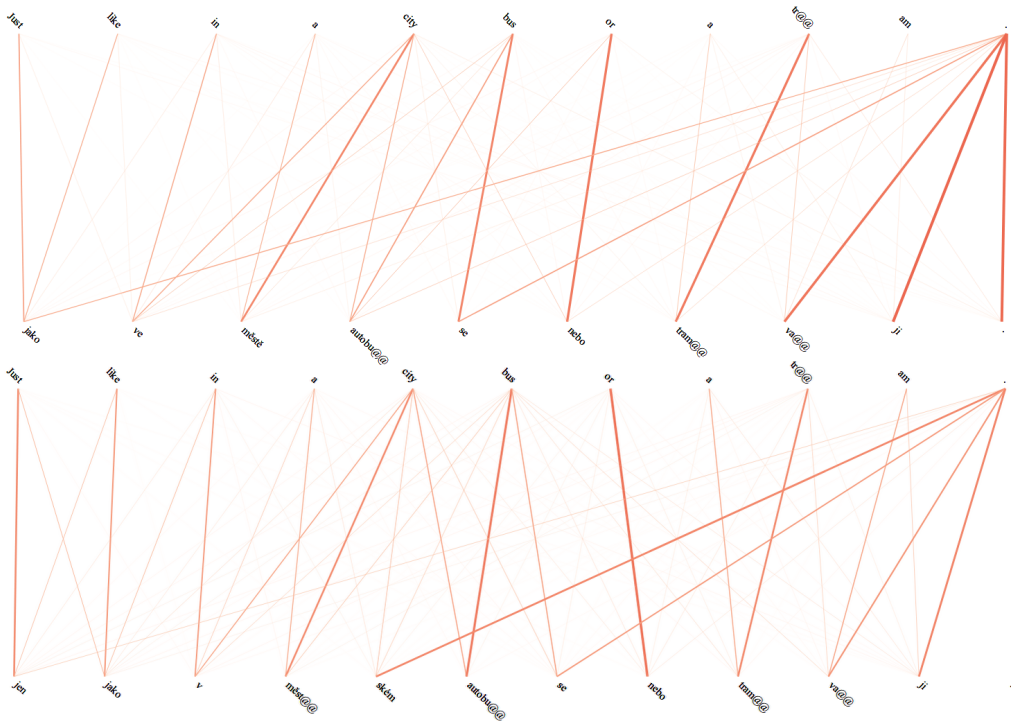
Figure 10: Fragment of soft alignments of the example sentence from the improved NMT system.

"nástěnné elektronické mapě" was not identified by the MWE extractor at all.

Figure 7 illustrates translations of an example sentence by the En→Lv NMT systems. The MWE, in this case, is "network users" that is translated as "tīkla lietotāji" by the modified system and completely mistranslated by the baseline.

4.4 Alignment Inspection

For inspecting the NMT attention alignments, we developed a tool (Rikters et al., 2017) that takes data produced by Neural Monkey—a 3D array (tensor) filled with the alignment probabilities together with source and target subword units (Sennrich et al., 2016b) or byte pair encodings (BPEs)—as input and produces a soft alignment matrix (Figure 8) of the subword units that highlights all units, that get attention when translating a specific subword unit. The tool includes a web version that was adapted from Nematus (Sennrich et al., 2017) utilities and slightly modified. It allows to output the soft alignments in a different perspective, as connections between BPEs as visible in Figure 9 and Figure 10.



Source:	Just like in a city bus or a tram.
Baseline:	Jako ve městě autobuse nebo tramvaji.
Improved NMT:	Jen jako v městském autobuse nebo tramvaji.
Reference:	Stejně jako v městském autobuse či tramvaji.

Figure 11: Soft alignment example visualizations from translating an English sentence into Czech from the baseline (top, hypothesis 1) and improved (bottom, hypothesis 2) NMT systems.

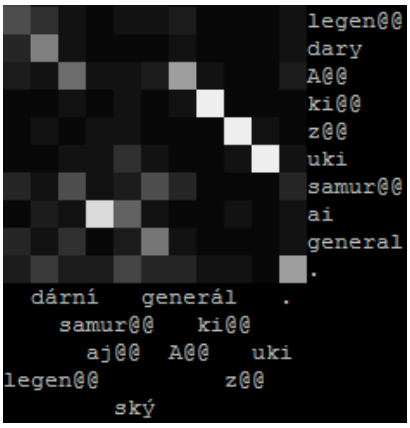


Figure 8: Example of a soft alignment matrix.

In these examples, the attention state of the previously mentioned MWE from En→Lv translations ("network users") is visible. The alignment inspection tool allows to see that the baseline NMT in Figure 9 has multiple faded alignment lines for both words "network" and "users", which outlines that the neural network is unsure and looking all around for traces to the correct translation. However, in Figure 10, it is visible that both these words have strong alignment lines to the words "tīkla lietotāji", that were also identified by the MWE Toolkit as an MWE candidate.

Figure 11 shows one of the previously mentioned En→Cs translation examples. Here it is clear that in the baseline alignment no attention goes to the word "městě" or the subword units "autobu@@" and "se" when translating "city". In the modified version, on the other hand, some attention from "city" goes into all closely related subword units: "měst@@", "ském", "autobu@@", and "se". It is also visible that in this example, the translation of "bus" gets attention from not only "autobu@@" and "se" but also the ending subword unit of "city", i.e. the token "ském".

5 Conclusion

In this paper, we described the first experiments with handling multi-word expressions in neural machine translation systems. Details on identifying and extracting MWEs from parallel corpora, as well as aligning them and building corpora of parallel MWEs were provided. We explored two methods of integrating MWEs in training data for NMT and examined the output translations of the trained NMT systems with custom built tools for alignment inspection.

In addition to the methods described in this paper, we also released open-source scripts for a complete workflow of identifying, extracting and integrating MWEs into the NMT training and translation workflow.

While the experiments did not show outstanding improvements on the general development data set, an increase of 0.99 BLEU was observed when using an MWE specific test data set. Manual inspection of the output translations confirmed that translations of specific MWEs were improving after populating the training data with synthetic MWE data.

As the next steps, we plan (1) to analyze the obtained results of our experiments in more detail through the help of a larger scale manual human evaluation of the NMT output and (2) to continue experiments to find best ways how to treat different categories of MWEs, i.e. idioms.

Acknowledgement

This study was supported in parts by the grants H2020-ICT-2014-1-645442 (QT21), the ICT COST Action IC1207 *ParseME: Parsing and multi-word expressions. Towards linguistic precision and computational efficiency in natural language processing*, and Charles University Research Programme "Progres" Q18 – Social Sciences: From Multidisciplinarity to Interdisciplinarity.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Baldwin, T. and Kim, S. N. (2010). Multiword expressions. In *Handbook of Natural Language Processing, Second Edition*, pages 267–292. Chapman and Hall/CRC.

Bojar, O., Helcl, J., Kocmi, T., Libovický, J., and Musil, T. (2017). Results of the WMT17

Neural MT Training Task. In *Proceedings of the Second Conference on Machine Translation (WMT17)*, Copenhagen, Denmark.

Bouamor, D., Semmar, N., and Zweigenbaum, P. (2012). Identifying bilingual multi-word expressions for statistical machine translation. In *LREC*, pages 674–679.

Chen, W., Matusov, E., Khadivi, S., and Peter, J.-T. (2016). Guided alignment training for topic-aware neural machine translation. *AMTA 2016, Vol.*, page 121.

Cohn, T., Hoang, C. D. V., Vymolova, E., Yao, K., Dyer, C., and Haffari, G. (2016). Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California. Association for Computational Linguistics.

Deksne, D., Skadins, R., and Skadina, I. (2008). Dictionary of multiword expressions for translation into highly inflected languages. In *LREC*.

Helcl, J. and Libovický, J. (2017). Neural Monkey: An open-source tool for sequence learning. *The Prague Bulletin of Mathematical Linguistics*, (107):5–17.

Madnani, N. (2011). ibleu: Interactively debugging and scoring statistical machine translation systems. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 213–214. IEEE.

Majchráková, D., Dušek, O., Hajič, J., Karčová, A., and Garabík, R. (2012). Semi-automatic detection of multiword expressions in the slovak dependency treebank.

Paikens, P., Rituma, L., and Pretkalnina, L. (2013). Morphological analysis with limited resources: Latvian example. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013); May 22-24; 2013; Oslo University; Norway. NEALT Proceedings Series 16*, number 085, pages 267–277. Linköping University Electronic Press.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pecina, P. (2008). Reference data for czech collocation extraction. In *Proc. of the LREC Workshop Towards a Shared Task for MWEs (MWE 2008)*, pages 11–14.

Pinnis, M. (2013). Context independent term mapper for european languages. In *RANLP*, pages 562–570.

Ramisch, C. (2012). A generic framework for multiword expressions treatment: from acquisition to applications. In *Proceedings of ACL 2012 Student Research Workshop*, pages 61–66. Association for Computational Linguistics.

Rikters, M., Fishel, M., and Bojar, O. (2017). Visualizing Neural Machine Translation Attention and Confidence. *The Prague Bulletin of Mathematical Linguistics*, 109.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli Barone, A. V., Mokry, J., and Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Skadiņa, I. (2016). Multi-word expressions in english-latvian. In *Human Language Technologies–The Baltic Perspective: Proceedings of the Seventh International Conference Baltic HLT 2016*, volume 289, page 97. IOS Press.

Tang, Y., Meng, F., Lu, Z., Li, H., and Yu, P. L. H. (2016). Neural machine translation with external phrase memory. *CoRR*, abs/1606.01792.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Enabling Multi-Source Neural Machine Translation By Concatenating Source Sentences In Multiple Languages

Raj Dabre raj@nlp.ist.i.kyoto-u.ac.jp
Graduate School of Informatics, Kyoto University, Kyoto, Japan

Fabien Cromieres fabien@pa.jst.jp
Japan Science and Technology Agency, Saitama, Japan

Sadao Kurohashi kuro@i.kyoto-u.ac.jp
Graduate School of Informatics, Kyoto University, Kyoto, Japan

Abstract

In this paper, we explore a simple solution to "Multi-Source Neural Machine Translation" (MSNMT) which only relies on preprocessing a N-way multilingual corpus without modifying the Neural Machine Translation (NMT) architecture or training procedure. We simply concatenate the source sentences to form a single long multi-source input sentence while keeping the target side sentence as it is and train an NMT system using this preprocessed corpus. We evaluate our method in resource poor as well as resource rich settings and show its effectiveness (up to 4 BLEU using 2 source languages and up to 6 BLEU using 5 source languages). We also compare against existing methods for MSNMT and show that our solution gives competitive results despite its simplicity. We also provide some insights on how the NMT system leverages multilingual information in such a scenario by visualizing attention.

1 Introduction

Multi-Source Machine Translation (MSMT) Och and Ney (2001) is an approach that allows one to leverage source sentences in multiple languages to improve the translations to a target language. Typically N-way (or N-lingual) corpora are used for MSMT. N-way corpora are those in which translations of the same sentence exist in N different languages. This setting is realistic and has many applications. For example, the European Parliament maintains its proceedings in 21 languages. In Spain, international news companies write news articles in English as well as Spanish. One can now utilize the same sentence written in two different languages like Spanish and English to translate to a third language like Italian by utilizing a large English-Spanish-Italian trilingual corpus.

Neural machine translation (NMT) Bahdanau et al. (2015); Cho et al. (2014); Sutskever et al. (2014) enables one to train an end-to-end system without the need to deal with word alignments, translation rules and complicated decoding algorithms, which are a characteristic of phrase based statistical machine translation (PBSMT) systems. However, it is reported that NMT works better than PBSMT only when there is an abundance of parallel corpora. In a low resource scenario, vanilla NMT is either worse than or comparable to PBSMT Zoph et al. (2016).

Multi-source Neural Machine translation (MSNMT) involves using NMT for MSMT. Two

major approaches for Multi-Source NMT (MSNMT) have been explored, namely the multi-encoder (ME/me) Zoph and Knight (2016) and multi-source ensembling (ENS/ens) Garmash and Monz (2016); Firat et al. (2016). The multi-encoder approach involves extending the vanilla NMT architecture to have an encoder for each source language leading to larger models. On the other hand, the ensembling approach is simpler since it involves training multiple bilingual NMT models each with a different source language but the same target language.
We have discovered that there is an even simpler way to do MSNMT. We explore a new simplified end-to-end method that avoids the need to modify the NMT architecture as well as the need to learn an ensemble function. We simply concatenate the source sentences leading to a parallel corpus where the source side is a long multilingual sentence and the target side is a single sentence which is the translation of the aforementioned multilingual sentence. This corpus is then fed to any NMT training pipeline whose output is a multi-source NMT model.
The main contributions of this paper are as follows:

- Exploring a simple preprocessing step that allows for Multi-Source NMT (MSNMT) without any change to the NMT architecture[1].
- An exhaustive study of how the approach works in a resource poor as well as a resource rich setting.
- An analysis of how gains in the translation quality are correlated with language similarity in a multi-source scenario.
- An empirical comparison of our approach against two existing methods Zoph and Knight (2016); Firat et al. (2016) for MSNMT.
- An analysis of how NMT gives more importance to certain linguistically closer languages while doing multi-source translation by visualizing attention vectors.

2 Related Work

One of the first studies on multi-source MT Och and Ney (2001) explored how word based SMT systems would benefit from multiple source languages. Although effective, it suffered from a number of limitations that classic word and phrase based SMT systems do including the inability to perform end-to-end training. The work on multi-encoder multi source NMT Zoph and Knight (2016) is the first multi-source NMT approach which focused on utilizing French and German as source languages to translate to English. However their method led to models with substantially larger parameter spaces and they did not experiment with many languages. Moreover, since the encoders for each source language are separate it is difficult to explore how the source languages contribute towards the improvement in translation quality. Multi-source ensembling using a multilingual multi-way NMT model Firat et al. (2016) is an end-to-end approach but requires training a very large and complex NMT model. The work on multi-source ensembling which uses separately trained single source models Garmash and Monz (2016) is comparatively simpler in the sense that one does not need to train additional NMT models but the approach is not truly end-to-end since it needs an ensemble function to be learned. This method also helps eliminates the need for N-way corpora which allows one to exploit bilingual corpora which are larger in size. In all cases one ends up with either one large model or many small models for which an ensemble function needs to be learned.
Other related works include Transfer Learning Zoph et al. (2016) and Zero Shot NMT Johnson et al. (2016) which help improve NMT performance for low resource languages. Finally it is important to note works that involve the creation of N-way corpora: United Nations (Ziemski et al. (2016)), Europarl (Koehn (2005)), Ted Talks (Cettolo et al. (2012)), ILCI (Jha (2010)) and Bible (Christodouloupoulos and Steedman (2015)) corpora.

[1] One additional benefit of our approach is that any NMT architecture can be used, be it attention based or hierarchical NMT.

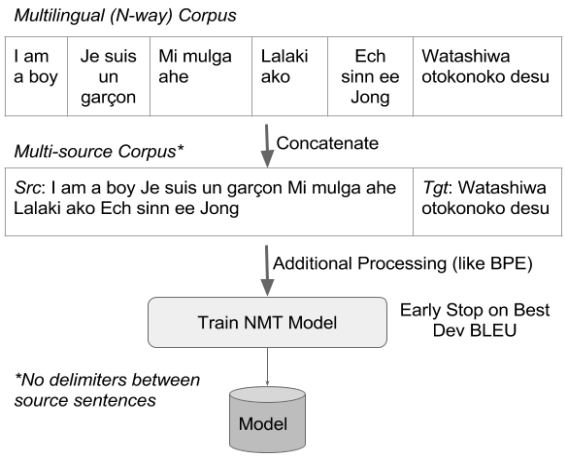


Figure 1: The Multi-Source NMT Approach We Explored.

3 Our Method

Refer to Figure 1 for an overview of our method which is as follows: For each target sentence **concatenate the corresponding source sentences** leading to a parallel corpus where the source sentence is a very long sentence that conveys the same meaning in multiple languages. An example line in such a corpus would be: source: “Hello Bonjour Namaskar Kamusta Hallo" and target:“konnichiwa". The 5 source languages here are English, French, Marathi, Filipino and Luxembourgish whereas the target language is Japanese. In this example each source sentence is a word conveying “Hello" in different languages. Note that there are **no delimiters between the individual source sentences** since we expect the NMT system will figure out the sentence boundaries by itself. We romanize the Marathi and Japanese words for readability. Optionally, one can perform additional processing, like Byte Pair Encoding (BPE), to overcome data sparsity and eliminate the unknown word rate. Use the training corpus to learn an NMT model using any off the shelf NMT toolkit. The order of the sentences belonging to different languages is kept the same in the training, development and test sets.

3.1 Other methods for comparison

3.1.1 Multi-Encoder Multi-Source Method

This method was proposed by Zoph and Knight (2016). The main idea is to have an encoder for each source language and concatenate encoding information before feeding it to the decoder. We use the technique where attentions are computed for both source languages and feed this multi-source attention to the decoder to predict a target word.

3.1.2 Multi-Source Ensembling Method

This method was proposed by Firat et al. (2016) and it relies on a single multilingual NMT model with separate encoders and decoders for each source and target language. All encoders and decoders share a single attention mechanism. To perform multi-source translation the model is fed source sentences in different languages and the logits are averaged (ensembling) before computing softmax to predict a target word. Since training a multilingual-multiway model is difficult and time consuming to train we rely on separately trained models for each source language and ensemble them without learning an ensemble function.

4 Experimental Settings

All of our experiments were performed using an encoder-decoder NMT system with attention for the various baselines and multi-source experiments. In order to enable infinite vocabulary and reduce data sparsity we use the Byte Pair Encoding (BPE) based word segmentation approach Sennrich et al. (2016). However we perform a slight modification to the original code where instead of specifying the number of merge operations manually we specify a desired vocabulary size and the BPE learning process automatically stops after it learns enough rules to obtain the prespecified vocabulary size. We prefer this approach since it allows us to learn a minimal model and it resembles the way Google's NMT system Wu et al. (2016) works with the Word Piece Model (WPM) Schuster and Nakajima (2012). We evaluate our models using the standard BLEU Papineni et al. (2002) metric[2] on the translations of the test set. Baseline models are single source models.

4.1 Languages and Corpora Settings

corpus type	Languages	train	dev2010	tst2010/tst2013
3 lingual	Fr, De, En	191381	880	1060/886
4 lingual	Fr, De, Ar, En	84301	880	1059/708
5 lingual	Fr, De, Ar, Cs, En	45684	461	1016/643

Table 1: Statistics for the the N-lingual corpora extracted from the IWSLT corpus for the languages French (Fr), German (De), Arabic (Ar), Czech (Cs) and English (En)

All of our experiments were performed using the publicly available ILCI[3] (Jha (2010)), United Nations[6] (Ziemski et al. (2016)) and IWSLT[7] (Cettolo et al. (2015)) corpora.
The ILCI corpus is a 6-way multilingual corpus spanning the languages Hindi, English, Tamil, Telugu, Marathi and Bengali was provided as a part of the task. The target language is Hindi and thus there are 5 source languages. The training, development and test sets contain 45600, 1000 and 2400 6-lingual sentences respectively[8]. Hindi, Bengali and Marathi are Indo-Aryan languages, Telugu and Tamil are Dravidian languages and English is a European language. In this group English is the farthest from Hindi, grammatically speaking, whereas Marathi is the closest to it. Morphologically speaking, Bengali is closer to Hindi compared to Marathi (which has agglutinative suffixes) but Marathi and Hindi share the same script and they also share more cognates compared to the other languages. It is natural to expect that translating from Bengali and Marathi to Hindi should give Hindi sentences of higher quality as compared to those obtained by translating from the other languages and thus using these two languages as source languages in multi-source approaches should lead to significant improvements in translation quality. We verify this hypothesis by exhaustively trying all language combinations.
The IWSLT corpus is a collection of 4 bilingual corpora spanning 5 languages where the target language is English: French-English (234992 lines), German-English (209772 lines), Czech-English (122382 lines) and Arabic-English (239818 lines). Linguistically speaking French and German are the closest to English followed by Czech and Arabic. In order to obtain N-lingual

[2]This is computed by the multi-bleu.pl script, which can be downloaded from the public implementation of Moses Koehn et al. (2007).

[3]This was used for the Indian Languages MT task in ICON 2014[4] and 2015[5].

[6]https://conferences.unite.un.org/uncorpus

[7]https://wit3.fbk.eu/mt.php?release=2016-01

[8]In the task there are 3 domains: health, tourism and general. However, we focus on the general domain in which half the corpus comes from the health domain the other half comes from the tourism domain.

sentences we only keep the sentence pairs from each corpus such that the English sentence is present in all the corpora. From the given training data we extract trilingual (French, German and English), 4-lingual (French, German, Arabic and English) and 5-lingual corpora. Similarly we extract 3, 4 and 5 lingual development and test sets. The IWSLT corpus (downloaded from the link given above) comes with a development set called dev2010 and test sets named tst2010 to tst2013 (one for each year from 2010 to 2013). Unfortunately only the tst2010 and tst2013 test sets are N-lingual. Refer to Table 1 which contains the number of lines of training, development and test sentences we extracted.
The UN corpus spans 6 languages: French, Spanish, Arabic, Chinese, Russian and English. Although there are 11 million 6-lingual sentences we use only 2 million for training since our purpose was not to train the best system but to show that our method works in a resource rich situation as well. The development and test sets provided contain 4000 lines each and are also available as 6-lingual sentences. We chose English to be the target language and focused on Spanish, French, Arabic and Russian as source languages. Due to lack of computational facilities we only worked with the following source language combinations: French and Spanish, French and Russian, French and Arabic and Russian and Arabic.

4.2 NMT Systems and Model Settings

For training various NMT systems, we used the open source KyotoNMT toolkit[9] Cromieres et al. (2016). KyotoNMT implements an Attention based Encoder-Decoder Bahdanau et al. (2015) with slight modifications to the training procedure. We modify the NMT implementation in KyotoNMT to enable multi encoder multi source NMT Zoph and Knight (2016). Since the NMT model architecture used in Zoph and Knight (2016) is slightly different from the one in KyotoNMT, the multi encoder implementation is not identical (but is equivalent) to the one in the original work. For the rest of the paper “baseline" systems indicate single source NMT models trained on bilingual corpora. We train and evaluate the following NMT models:

- One source to one target.
- N source to one target using our proposed multi source approach.
- N source to one target using the multi encoder multi source approach Zoph and Knight (2016).
- N source to one target using the multi source ensembling approach that late averages[10] Firat et al. (2016) N one source to one target models[11].

  The model and training details are as follows:
- BPE vocabulary size: 8k[12] (separate models for source and target) for ILCI and IWSLT corpora settings and 16k for the UN corpus setting. When training the BPE model for the source languages we learn a single shared BPE model. In case of languages that use the same script it allows for cognate sharing thereby reducing the overall vocabulary size.
- Embeddings: 620 nodes
- RNN (Recurrent Neural Network) for encoders and decoders: LSTM with 1 layer, 1000 nodes output. Each encoder is a bidirectional RNN.
- In the case of multiple encoders, one for each language, each encoder has its own separate vocabulary.
- Attention: 500 nodes hidden layer. In case of the multi encoder approach there is a separate attention mechanism per encoder.

[9] https://github.com/fabiencro/knmt

[10] Late averaging implies averaging the logits of multiple decoders before computing softmax to predict the target word.

[11] In the original work a single multilingual multiway NMT model was trained and ensembled but we train separate NMT models for each source language.

[12] We also try vocabularies of size 16k and 32k but they take longer to train and overfit badly in a low resource setting

- Batch size: 64 for single source, 16 for 2 sources and 8 for 3 sources and above for IWSLT and ILCI corpora settings. 32 for single source and 16 for 2 sources for the UN corpus setting.
- Training steps: 10k[13] for 1 source, 15k for 2 source and 40k for 5 source settings when using the IWSLT and ILCI corpora. 200k for 1 source and 400k for 2 source for the UN corpus setting to ensure that in both cases the models get saturated with respect to heir learning capacity.
- Optimization algorithms: Adam with an initial learning rate of 0.01
- Choosing the best model: Evaluate the model on the development set and select the one with the best BLEU Papineni et al. (2002) after reversing the BPE segmentation on the output of the NMT model.
- Beam size for decoding: 16[14]

We train and evaluate the following NMT models using the ILCI corpus:

- One source to one target: 5 models (Baselines)
- Two source to one target: 10 models (5 source languages, choose 2 at a time)
- Five source to one target: 1 model

In this setting, we also calculate the ***lexical similarity***[15] between the languages involved in using the Indic NLP Library[16]. The objective behind this is to determine whether or not lexical similarity, which is also one of the indicators of linguistic similarity and hence translation quality Kunchukuttan and Bhattacharyya (2016), is also an indicator of how well two source languages work together.

In the IWSLT corpus setting we did not try various combinations of source languages as we did in the ILCI corpus setting. We train and evaluate the following NMT models for each N-lingual corpus:

- One source to one target: N-1 models (Baselines; 2 for the trilingual corpus, 3 for the 4-lingual corpus and 4 for the 5-lingual corpus)
- N-1 source to one target: 3 models (1 for trilingual, 1 for 4-lingual and 1 for 5-lingual)

Similarly for the UN corpus setting we only tried the following one source one target models: French-English, Russian-English, Spanish-English and Arabic-English. The two source combinations we tried were: French+Spanish, French+Arabic, French+Russian, Russian+Arabic. The target language is English.

For the ILCI corpus setting, Table 2 contains the BLEU scores for all the settings and lexical similarity scores for all combinations of source languages, two at a time. The caption contains a complete description of the table. The last row of Table 2 contains the BLEU score for all the multi source settings which uses all 5 source languages.
For the results of the IWSLT corpus setting, refer to Table 3. Finally, refer to Table 4 for the UN corpus setting.

4.3 Analysis

4.3.1 Main findings

From Tables 2, 3 and Table 4 it is clear that our simple source sentence concatenation based approach (under columns labeled “our") is able to leverage multiple languages leading to significant improvements compared to the BLEU scores obtained using any of the individual

[13] We observed that the models start overfitting around 7k-8k iterations

[14] We performed evaluation using beam sizes 4, 8, 12 and 16 but found that the differences in BLEU between beam sizes 12 and 16 are small and gains in BLEU for beam sizes beyond 16 are insignificant

[15] https://en.wikipedia.org/wiki/Lexical_similarity

[16] http://anoopkunchukuttan.github.io/indic_nlp_library

Source Language 1	Source Language 2 [XX-Hi BLEU] *XX-Hi sim*															
	En [11.08] *0.20*				**Mr** [24.60] *0.51*				**Ta** [10.37] *0.30*				**Te** [16.55] *0.42*			
	our	ens	me	*sim*	our	ens	me	*sim*	our	ens	me	*sim*	our	ens	me	*sim*
Bn [19.14] *0.52*	**20.70**	19.45	19.10	*0.18*	29.02	**30.10**	27.33	*0.46*	19.85	**20.79**	18.26	*0.30*	22.73	**24.83**	22.14	*0.39*
En [11.08] *0.20*	-				25.56	23.06	**26.01**	*0.20*	14.03	**15.05**	13.30	*0.18*	18.91	**19.68**	17.53	*0.20*
Mr [24.60] *0.51*	-				-				**25.64**	24.70	23.79	*0.33*	27.62	**28.00**	26.63	*0.43*
Ta [10.37] *0.30*	-				-				-				18.14	**19.11**	17.34	*0.38*
All	our: **31.56**				ens: 30.29				me: 28.31							

Table 2: **ILCI corpus results**: BLEU scores for two source to one target setting for all language combinations and for five source to one target using the ILCI corpus. The languages are Bengali (Bn), English (En), Marathi (Mr), Tamil (Ta), Telugu (Te) and Hindi (Hi). Each language is accompanied by the BLEU score for translating to Hindi from that language and its lexical similarity with Hindi. Each cell in the upper right triangle contains the BLEU scores using a. Our proposed approach (our), b. Multi source ensembling approach (ens), c. Multi Encoder Multi Source approach (me) and d. The lexical similarity (sim; in tiny font size). The best BLEU score is in bold. The train, dev, test split sizes are 45600, 1000 and 2400 lines respectively.

source languages. The ensembling (under columns labeled “ens") and the multi encoder (under columns labeled “me") approaches also lead to improvements in BLEU. Note that in every single case, gains in BLEU are statistically significant regardless of the methods used. It should be noted that in a resource poor scenario ensembling generally outperforms all other approaches but in a resource rich scenario our method as well as the multi encoder method are much better. However, the comparison with the ensembling method is unfair to our method since the former uses N times more parameters than the latter. However, one important aspect of our approach is that the model size for the multi source systems is the same as that of the single source systems since the vocabulary sizes are exactly the same. The multi encoder systems involve more parameters whereas the ensembling approach does not allow for the source languages to truly interact with each other.

4.3.2 Correlation between linguistic similarity and gains using multiple sources

In the case of the ILCI corpus setting, Table 2, it is clear that no matter which source languages are combined, the BLEU scores are higher than those given by the single source systems. Marathi and Bengali are the closest to Hindi (linguistically speaking) compared to the other languages and thus when used together they help obtain an improvement of 4.39 BLEU points compared to when Marathi is used as the only source language (24.63). However it can be seen that combining any of Marathi, Bengali and Telugu with either English or Tamil lead to smaller gains. There is a strong correlation between the gains in BLEU and the lexical similarity. Bengali and English which have the least lexical similarity (0.18) give only a 1.56 BLEU improvement whereas Bengali and Marathi which have the highest lexical similarity (0.46) give a BLEU improvement of 4.42 using our multi-source method. This seems to indicate that although multiple source languages do help, source languages that are linguistically closer to each other are responsible for maximum gains (as evidenced by the correlation between lexical similarity and gains in BLEU). Finally, the last row of Table 2 shows that using additional languages lead to further gains leading to a BLEU score of 31.56 which is 6.96 points above when only Marathi is used as the only source language and 2.54 points above when Marathi and Bengali are used as the source languages. As future work it will be worthwhile to investigate the diminishing returns in BLEU improvement obtained per additional language.

Corpus Type *Train Size*	Language Pair	BLEU tst2010	BLEU tst2013	Number of sources	BLEU tst2010			BLEU tst2013		
					our	ens	me	our	ens	me
3 lingual *191381 lines*	**Fr-En**	19.72	22.05	**2**	**22.56**	18.64	22.03	**24.02**	18.45	23.92
	De-En	16.19	16.13							
4 lingual *84301 lines*	**Fr-En**	9.02	7.78	**3**	11.70	**12.86**	10.30	9.16	**9.48**	7.30
	De-En	7.58	5.45							
	Ar-En	6.53	5.25							
5 lingual *45684 lines*	**Fr-En**	6.69	6.36	**4**	8.34	**9.23**	7.79	**6.67**	6.49	5.92
	De-En	5.76	3.86							
	Ar-En	4.53	2.92							
	Cs-En	4.56	3.40							

Table 3: **IWSLT corpus results**: BLEU scores for the single source and N source settings using the IWSLT corpus. The languages are French (Fr), German (De), Arabic (Ar), Czech (Cs) and English (En). We give the BLEU scores for two test sets tst2010 and tst2013 which we translate using a. Our proposed approach (our), b. Multi source ensembling approach (ens) and c. Multi Encoder Multi Source approach (me). The best BLEU score is in bold. The train corpus sizes are given in tiny font size. Refer to Table 1 for details on corpora sizes.

Language Pair	BLEU
Es-En	49.20
Fr-En	40.52
Ar-En	40.58
Ru-En	38.94

Source Combination	BLEU		
	our	ens	me
Es+Fr	**49.93***	46.65	47.39
Fr+Ru	**43.99**	40.63	42.12
Fr+Ar	43.85	41.13	**44.06**
Ar+Ru	41.66	43.12	**43.69**

Table 4: **UN corpus results**: BLEU scores for the single source and 2 source settings using the UN corpus. The languages are Spanish (Es), French (Fr), Russian (Ru), Arabic (Ar) and English (En). We give the BLEU scores for for the test set which we translate using a. Our proposed approach (our), b. Multi source ensembling approach (ens) and c. Multi Encoder Multi Source approach (me). Note that we do not try all language pairs. The highest score is the one in bold. All BLEU score improvements are statistically significant ($p < 0.001$) compared to those obtained using either of the source languages independently. The train, dev, test split sizes are 2M, 4k and 4k lines respectively.

4.3.3 Performance in resource rich settings

In the UN corpus setting, Table 4, where we used approximately 2 million training sentences, we also obtained improvements in BLEU. In the case of the single source systems we observed that the BLEU score for Spanish-English was around 9 BLEU points higher than for French-English which is consistent with the observations in the original work concerning the construction of the UN corpus Ziemski et al. (2016). Furthermore, combining using French and Spanish together leads to a small (0.7) improvement in BLEU (over Spanish-English) that is statistically significant ($p < 0.001$) which is to be expected since the BLEU for Spanish-English is already much better than the BLEU for French-English. Since the BLEU scores for French, Arabic and Russian to English are closer to each other we can see that the BLEU scores for French+Arabic, French+Russian and Arabic+Russian to English are around 3 BLEU points higher than those of

their respective single source counterparts.However, they do not beat the performance[17] of the multi-encoder models which have roughly twice the number of parameters.
Similar gains in BLEU are observed in the IWSLT corpus setting. Halving the size of the training corpus (from trilingual to 4-lingual) leads to baseline BLEU scores being reduced by half (19.72 to 9.62 for French-English tst2010 test set) but using an additional source leads to a gain of roughly 2 BLEU points. Although the gains are not as high as seen in the ILCI corpus setting it must be noted that the test set for the ILCI corpus is easier in the sense that it contains many short sentences compared to the IWSLT test sets. Our method does not show any gains in BLEU for the tst2013 test set in the 4-lingual setting, an anomaly which we plan to investigate in the future.

4.4 Studying multi-source attention

In order to understand whether or not our multi-source NMT approach prefers certain language over others, we obtained visualizations for the attention vectors for a few sentences from the test set. Refer to Figure 2 for an example. Firstly, it can be seen that the NMT model learns sentence boundaries although we did not specify delimiters between sentences, Note that, in the figure, we use a horizontal line to separate the languages but the NMT system receives a single, long multi-source sentence. The words of the target sentence in Hindi are arranged from left to right along the columns whereas the words of the multi-source sentence are arranged from top to bottom across the rows. Note that the source languages (and lexical similarity scores with Hindi) are in the following order: Bengali (0.52), English (0.20), Marathi (0.51), Tamil (0.30), Telugu (0.42).
The most interesting thing that can be seen is that the attention mechanism focuses on each language but with varying degrees of focus. Bengali, Marathi and Telugu are the three languages that receive most of the attention (highest lexical similarity scores with Hindi) whereas English and Tamil (lowest lexical similarity scores with Hindi) barely receive any. Building on this observation we believe that the gains we obtained by using all 5 source languages were mostly due to Bengali, Telugu and Marathi whereas the NMT system learns to practically ignore Tamil and English. However there does not seem to be any detrimental effect of using English and Tamil.
From Figure 3 it can be seen that this observation also holds in the UN corpus setting for French+Spanish to English where the attention mechanism gives a higher weight to Spanish words compared to French words since the Spanish-English translation quality is about 9 BLEU points higher than the French-English translation quality. It is also interesting to note that the attention can potentially be used to extract a multilingual dictionary simply by learning a N-source NMT system and then generating a dictionary by extracting the words from the source sentence that receive the highest attention for each target word generated.

5 Conclusion and Future Work

In this paper, we have explored a simple approach for “Multi-Source Neural Machine Translation" that can used with any NMT system seen as a black-box. We have evaluated it in a resource poor as well as a resource rich setting using the ILCI, IWSLT and UN corpora. We have compared our approach with two other previously proposed approaches and showed that it gives competitive results with other state of the art methods while using less than half the number of parameters (for 2 source models). It is domain and language independent and the gains are significant. We also observed, by visualizing attention, that NMT is able to identify sentence boundaries without sentence delimiters and focuses on some languages by practically

[17]The difference in performance between multi-encoder approach and our approache for French+Arabic is not significant.

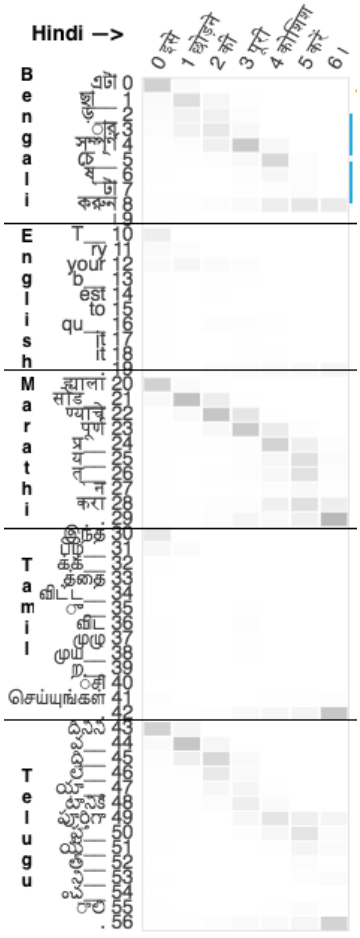


Figure 2: Attention Visualization for ILCI corpus setting for Bengali, English, Marathi, Tamil and Telugu to Hindi. A horizontal black line is used to separate the source languages but the NMT system receives a single, long multi-source sentence.

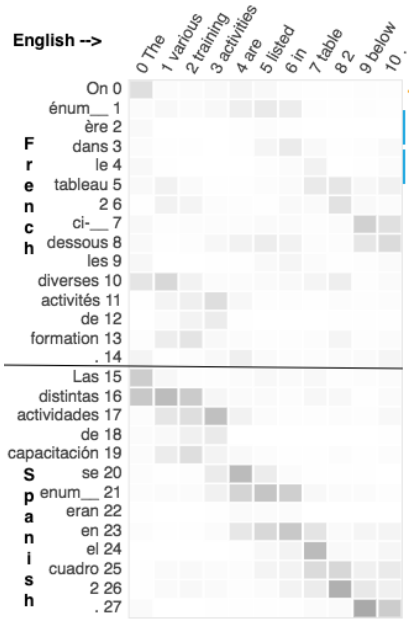


Figure 3: Attention Visualization for UN corpus setting for French and Spanish to English. A horizontal black line is used to separate the source languages but the NMT system receives a single, long multi-source sentence.

ignoring others indicating that language relatedness is one of the aspects that should be considered in a multilingual MT scenario. Although we have not explored other multi-source NLP tasks in this paper, we believe that our method and findings will be applicable to them.

In the future we plan on exploring the language relatedness phenomenon by considering even more languages. We also plan on investigating the extraction of multilingual dictionaries by analyzing the attention links and on how we can obtain a single NMT model that can translate up to N source languages and thereby function in a situation where some source sentences in certain languages are missing.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, USA. International Conference on Learning Representations.

Cettolo, M., Girardi, C., and Federico, M. (2012). Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Cattoni, R., and Federico, M. (2015). The iwslt 2015 evaluation campaign. In *Proceedings of the Twelfth International Workshop on Spoken Language Translation (IWSLT)*.

Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Christodouloupoulos, C. and Steedman, M. (2015). A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.

Cromieres, F., Chu, C., Nakazawa, T., and Kurohashi, S. (2016). Kyoto university participation to wat 2016. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 166–174, Osaka, Japan. The COLING 2016 Organizing Committee.

Firat, O., Sankaran, B., Al-Onaizan, Y., Yarman-Vural, F. T., and Cho, K. (2016). Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 268–277.

Garmash, E. and Monz, C. (2016). Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1409–1418, Osaka, Japan. The COLING 2016 Organizing Committee.

Jha, G. N. (2010). The tdil program and the indian langauge corpora intitiative (ilci). In Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F. B., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2016). Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *ACL*. The Association for Computer Linguistics.

Kunchukuttan, A. and Bhattacharyya, P. (2016). Orthographic syllable as basic unit for SMT between related languages. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1912–1917.

Och, F. J. and Ney, H. (2001). Statistical multi-source translation. In *Proceedings of MT Summit*, volume 8, pages 253–258.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Schuster, M. and Nakajima, K. (2012). Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 5149–5152.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Ziemski, M., Junczys-Dowmunt, M., and Pouliquen, B. (2016). The united nations parallel corpus v1.0. In Chair), N. C. C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Zoph, B. and Knight, K. (2016). Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.

Zoph, B., Yuret, D., May, J., and Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1568–1575.

Learning an Interactive Attention Policy for Neural Machine Translation

Samee Ibraheem* sibraheem@berkeley.edu
Nicholas Altieri* naltieri@berkeley.edu
John DeNero denero@berkeley.edu
**Equal contribution.*

Abstract

Interactive machine translation research has focused primarily on predictive typing, which requires a human to type parts of the translation. This paper explores an interactive setting in which humans guide the attention of a neural machine translation system in a manner that requires no text entry at all. The system generates a translation from left to right, but waits periodically for a human to select the word in the source sentence to be translated next. A central technical challenge is that the system must learn when and how often to request guidance from the human. These decisions allow the system to trade off translation speed and accuracy. We cast these decisions as a reinforcement learning task and develop a policy gradient approach to train the system. Critically, the system can be trained on parallel data alone by simulating human guidance at training time. Our experiments demonstrate the viability of this interactive setting to improve translation quality and show that an effective policy for periodically requesting human guidance can be learned automatically.

1 Introduction

Despite rapid advances in neural machine translation, human input is still needed to meet the translation quality requirements of many applications. Interactive machine translation seeks to combine the quality of human translation with the speed and lexical coverage of machine translation. This paper explores an interactive setting in which the human translator does not type at all, but instead guides the attention of a neural machine translation system by selecting relevant source words as the system translates. While we should not expect that the resulting translations will be as accurate as those produced by predictive typing, this interactive approach could provide fast and accurate draft translations that could later be improved by post-editing. Moreover, source word selection enables new user interface options because it can be performed using a wide variety of input devices, including a mouse, a touch screen, or an eye tracker, which may be used in tandem with traditional text entry methods.

We first address the question of whether guiding the attention of a neural machine translation system can provide enough useful information to improve translation quality. Rather than experimenting directly with human subjects, we compute an experimental upper bound on the accuracy gains from guided attention. For each word that the system is meant to generate, we find an oracle attention that maximizes the probability of generating that word. We find that guiding attention toward this oracle provides a great deal of information to the translation system, yielding substantial gains in translation quality.

Second, we define an interactive translation process in which the system generates a translation left-to-right, but pauses on occasion to request guidance from a human collaborator. Ide-

ally, the system would not pause after every word; if the system can generate some portion of the translation accurately without human intervention, then it would be wasteful for it to solicit human input. Therefore, an ideal system must learn to trade off between translating accurately and requiring as little human input as possible.

However, it is difficult to predict the long-term consequences of choosing whether or not to pause at any given position. The value of receiving human guidance is not only that it may improve the prediction of the next word, but that it may improve predictions of all subsequent words. Therefore, pausing early for human input might allow the system to require less guidance in later parts of a sentence. Our primary technical contribution is to cast the sequence of decisions about when to request human guidance as a reinforcement learning problem that properly accounts for the system's uncertainty about all the downstream effects of requesting human intervention. We apply a policy gradient method to this problem and show that the system is able to learn an effective interaction policy. This policy estimates when, during the process of translation, human guidance is likely to provide enough long-term benefit to justify the cost of pausing.

We evaluate our approach using an English-German neural machine translation system trained for the WMT 2016 news translation task. We show that the whole system, including the learned interaction policy, can be trained fully automatically by approximating human input using simulated guidance.

2 Related Work

Interactive machine translation involves human translators working collaboratively with a machine translation system to produce high quality output efficiently (Foster and Lapalme, 2002). Several interactive interfaces to machine translation systems have been designed and evaluated in the research community, such as TransType (Langlais et al., 2000), Thot (Ortiz-Martínez et al., 2010), and Caitra (Koehn, 2009). Green et al. (2014) investigates the trade-off between human effort and translation quality within the paradigms of post-editing and interactive MT.

A growing line of research has explored the use of neural machine translation with attention (Bahdanau et al., 2014) in an interactive setting. Wuebker et al. (2016) compares the performance of neural and statistical machine translation models for interactive prediction, and shows that neural models are substantially more accurate. Knowles and Koehn (2016) also demonstrates that neural models provide more accurate interactive predictions than statistical models and addresses efficiency challenges. Hokamp and Liu (2017) describes a search algorithm for neural models that specifically targets a typical interactive workflow in which the terms in a bilingual lexicon must be prioritized over alternatives.

Werling et al. (2015) investigates the trade-off between the cost of human intervention and accuracy for three other tasks: named-entity recognition, sentiment classification, and image classification. That work also proposes an approach to decision making that considers the uncertain long-term consequences of actions.

Mi et al. (2016) demonstrates the usefulness of providing additional attention information to a fully automated neural machine translation system. In this work, the authors add an additional loss to the translation model which encourages the attention computed by the NMT system to resemble alignments predicted by an IBM word alignment model.

3 Guided Attention

Neural machine translation with attention (Bahdanau et al., 2014) is a variant of the seq2seq model (Sutskever et al., 2014) that incorporates attention over the source encodings into the decoder. The attention is a distribution over source positions that can be interpreted as a soft indicator of what part of the source sentence will be translated next. We propose to replace the

attention predicted by the model with a *guided* attention distribution that is provided directly by a human selecting a source word. In this paper, we simulate the human selection using the source word that is most helpful in the translation decision, described in detail below.

3.1 Neural Machine Translation with Attention

Given a source sentence $x = x_1, \ldots, x_n$ and a target sentence $y = y_1, \ldots, y_m$, the model first encodes x to form input representations $z_1, \ldots, z_n$. To predict the target labels y, the model conditions on a concatenation of two vectors, one being a hidden representation of the output generated so far, and the other being the input representations weighted by the attention: $\sum_i \alpha_i^{(t)} z_i$, where $\alpha_i^{(t)}$ is the attention computed at time t for the ith word in the source sentence. The input representations and hidden decoder states can be defined using an LSTM (Bahdanau et al., 2014) or convolution (Gehring et al., 2017) over word embeddings.

The attention vector is a distribution over source positions: $\sum_i \alpha_i^{(t)} = 1$ and $\alpha_i^{(t)} \geq 0$. To compute $\alpha_i^{(t)}$, a feed-forward neural network is used that takes in as inputs (z_i, h_t) where h_t is the hidden decoder state at time t. Finally, given the attention, hidden decoder state, and input representations, the label y_t is predicted using a learned distribution $p(y_t|h_t, \sum_i \alpha_i^{(t)} z_i)$.

3.2 Simulated Attention

Instead of using human input to train the model, we attempt to simulate the behavior of an accurate human, allowing for faster and cheaper training. We do this by, at each time step, calculating the distribution over the target vocabulary $p(y_t|h_t, z_i)$ for each i, which is equivalent to evaluating a one-hot attention vector for each source sentence word. We then provide the one-hot attention for the source word that had the highest predictive probability for the correct next target word to be translated. That is, if $i^* = \arg\max_i p(y^*|h_j, z_i)$, where y^* is the correct target word, then

$$\alpha^{(t)} = e_{i^*} \implies \sum_i \alpha_i^{(t)} z_i = z_{i^*}.$$

4 Learning When to Ask for Guidance

Given that we have a method for simulating the guidance that a human would provide, we turn to the problem of deciding when to request guidance at all. Each request for guidance affects the input representation used for predicting a single word. Over the course of a sentence, the system can request guidance multiple times.

4.1 Interaction Policy

To implement our interactive method, we use a greedy decoder. For each predicted word, the model decides whether to translate using guided attention or to translate using the attention predicted by the model. At the end of each iteration, there will be a loss penalty corresponding to the amount of guidance requested as well as the likelihood of the sentence under the model. Guidance improves likelihood by providing more information to each decision, but incurs a penalty for requesting guidance.

4.2 Interactive Machine Translation as Reinforcement Learning

We believe that reinforcement learning is an appropriate framework for our set up, since deciding when to ask for assistance can have long term ramifications on final accuracy that are hard to anticipate before training. We therefore model our framework by a Markov decision process (MDP). In this MDP, our agent is the machine translation system, whose actions are whether

or not to request guided attention, and our reward function is the cross-entropy between our prediction of the next word and a distribution that predicts the reference with probability 1.

4.3 Reinforcement Learning

An MDP is a tuple (S, A, T, R). S is the set of all possible states that an agent can be in. A is the set of all possible actions the agent can take. T is the transition function $p(s_{t+1}|s_t, a_t) = T(s_{t+1}|s_t, a_t)$ that is the distribution over the next state given the current state and the action to be taken. Finally, R is the reward function $R(s_{t+1}, a_t, s_t)$ that determines the reward for transitioning into s_{t+1} from s_t with action a_t.

An agent acting in a MDP can be described by a policy function $\pi : S \to A$, that takes in states and returns actions. It is the goal of reinforcement learning to learn a policy that maximizes the expected sum of (discounted) rewards: $\mathbf{E}[\sum_t \gamma^t R(s_{t+1}, \pi(s_t), s_t)]$, where $\gamma \in (0, 1]$ is the discount factor.

In the case of interactive attention in machine translation, a state s_t captures the activation of the translation network just before it would generate the next target word w_t. There are only two possible actions: whether to go ahead and generate w_t or to request guidance. If guidance is requested, then a new activation of the translation network is computed by replacing the model's attention weights with the guide's attention weights, and then a new word w'_t is generated using these new activations. If guidance is not requested, then w_t is generated. In either case, the reward function is the cross entropy sequence loss of the correct translation.

4.3.1 Policy Gradient

Policy gradient is a common reinforcement learning method to learn a policy π_θ parameterized by θ. The policy gradient method aims to perform stochastic gradient ascent on the objective

$$J(\theta) = \mathbf{E}\left[\sum_{t=1}^{T-1} \gamma^t R(s_{t+1}, \pi_\theta(s_t), s_t)\right].$$

Let $\pi_\theta(a_t|s_t)$ be the probability of choosing an action a_t in state s_t according to the policy π_θ. The policy gradient theorem states that if a_t are sampled according to $\pi_\theta(s_t)$, and s_{t+1} are sampled according to $T(\cdot|s_t, a_t)$, then an unbiased estimator of $\nabla_\theta J(\theta)$ is

$$\sum_{t=1}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \sum_{\tau=t}^{T-1} \gamma^{(\tau-t)} R(s_{\tau+1}, a_\tau, s_\tau).$$

Although using the above expression is an unbiased estimator, it can have high variance, prompting the use of variance reduction methods. For any function $b(s)$, the following is also an unbiased estimator:

$$\sum_{t=1}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \sum_{\tau=t}^{T-1} \gamma^{(\tau-t)} (R(s_{\tau+1}, a_\tau, s_\tau) - b(s_\tau)),$$

And the choice that minimizes variance is

$$b(s_\tau) = \mathbf{E}_{\pi_\theta}\left[\sum_{\tau=t}^{T-1} \gamma^{(\tau-t)} R(s_{\tau+1}, a_\tau, s_\tau)\right].$$

This optimal $b(s_\tau)$ can be approximated by a parameterized function V_ϕ, where we learn V_ϕ by approximately minimizing

$$\mathbf{E}_{\pi_\theta}\left(V_\phi(s_t) - \sum_{\tau=t}^{T-1} \gamma^{(\tau-t)} R(s_{\tau+1}, a_\tau, s_\tau)\right)^2.$$

Finally, a policy gradient algorithm alternates between taking a step of stochastic gradient ascent on $J(\theta)$ and taking multiple gradient steps on V_ϕ.

When using the approximate value function V_ϕ to reduce variance, the inner expression of the gradient is typically called the *advantage function* and denoted $A(s_t)$:

$$A(s_t) = \sum_{\tau=t}^{T-1} \left[\gamma^{(\tau-t)} R(s_{\tau+1}, a_\tau, s_\tau)\right] - V_\phi(s_t).$$

For our value function, we use a feed-forward neural network with two hidden layers of 32 units each, and for our policy function we use a neural network with one 32-unit hidden layer. The input to the former is the standard decoder inputs, which consist of the previously output token and the weighted sum of the hidden representations $\sum_i \alpha_i^{(t)} z_i$. The input to the latter additionally includes the original softmax layer input.

4.4 Action Frequency Regularization

Since our goals are to maximize translation accuracy while minimizing the number of times a human would have to intervene, we introduce an action weight parameter w_a, in order to manage the trade-off between accuracy and human effort. To promote accuracy during training, we have part of the reward at time step t be the negative cross entropy of the predictions at time t. To incorporate the number of times that the system requests guidance, we include not only the probability of requesting guidance, but also whether or not guidance was requested. In addition to these, we incorporate a threshold parameter ρ_a, to ensure that the action probabilities do not exceed the designated value. We thus use the following policy gradient objective:

$$\hat{A}(s_t) \cdot \log p_\theta(a_t) + w_a \cdot \max(0, p(a_t) - \rho_a) \cdot a_t,$$

where a_i is a binary scalar that takes value 1 if guidance was requested, and 0 otherwise, and $\hat{A}$ is the standardized advantage function.

That is,

$$\hat{A}(s_t) = \frac{A(s_t) - \mu(A(s_t))}{\sigma(A(s_t))}.$$

5 Experiments

We evaluate our model on the task of translating from English to German. Specifically, we first train a sequence-to-sequence model with attention, and then continue training using our reinforcement learning model. The baseline neural machine translation model was trained for 508,387 iterations.

5.1 Datasets

We use the English-German WMT 2016 news task dataset, which contains 4.2 million training sentence pairs. We apply BPE with 32,000 merge operations.

5.2 Architecture Details

For our base NMT system, we used Google's large seq2seq system implementation (Britz et al., 2017). For the encoder, we had 512 hidden units. For the decoder, both the GRU and the

attention have 512 units. [1]

5.3 Results

We evaluate our approach on all 3000 sentences of the WMT 2016 news-test2013 development set. We first evaluate the baseline fully automatic NMT model, which yields a BLEU (Papineni et al., 2002) score of 19.37. In comparison, our model which asks for guidance with a 100% probability has a BLEU score of 32.51. Thus, requesting guidance indeed improves translation quality for this model. However, requesting guidance for every word would require maximal human effort, as the human translator would be required to click at each time step.

We also evaluate a variety of learned policies on the same data and using the same baseline model. During policy learning, the parameters of the translation model are frozen, and only the parameters of the policy and value functions are learned. Varying the action weight and threshold values yields various guidance frequencies and corresponding BLEU scores. To determine whether the learned policy is requesting guidance efficiently, for each trained policy we also evaluate a random policy that asks for guidance with the same frequency as the reinforcement learning policy (Figure 5.3). The learned policy was able to achieve a BLEU score of 27.25 with observed guidance of about 54%, which improved upon the random policy by almost 2 BLEU and upon the baseline model by about 8 BLEU.

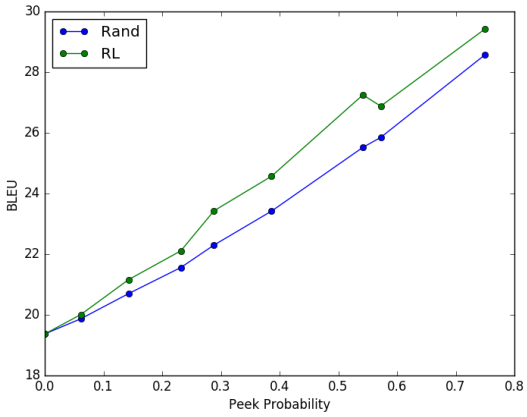


Figure 1: Translation accuracy for a random policy (blue) and a learned policy (green), for different guidance frequencies. More guidance provides higher accuracy. Across a range of guidance frequencies, the learned policy outperforms a policy that makes the same number of guidance requests, but at randomly chosen times.

6 Analysis

We compare the simulated clicks to the attention generated by the neural machine translator. In order to compare them, we compare the optimal word attention location computed by our simulator against the word with the largest weight according to the NMT system. This does provide a problem if the NMT was attending primarily to more than a single object, but nevertheless we believe this method of comparison may still provide useful intuition. In the figure below we

[1]For full specification see: `https://github.com/google/seq2seq/blob/master/example_configs/nmt_large.yml`

only include arrows for which the attended words differ. We note that using the simulated attention seems mostly intuitive with respect to where a human translator would click and corrects some of the NMT system errors. In particular, it makes *von* point to *of* and *zu* point to *counter*. However, there are also a few quirks. For example, it makes *kanishe* point to *Republic* and EOS point to *to*.

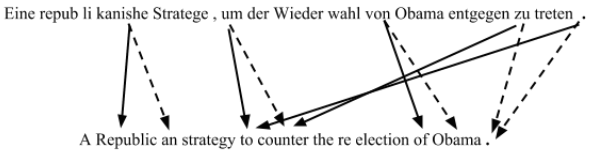


Figure 2: Guided attention (solid) vs NMT attention (dashed)

7 Future work

Our experiments demonstrate that reinforcement learning is an effective framework for requesting human guidance in interactive machine translation. However, we can identify several open questions that merit further investigation. First, we have focused on greedy decoding in this paper, because it is not trivial to apply a more sophisticated search procedure on top of our method. Developing an extension that incorporates beam search could improve performance. Second, during baseline training, the attention mechanism sees soft attention over the entire sentence as opposed to one hot attention over a single word, and the discrepancy between training and testing may limit the performance of the system. In addition, this method assumes that the word that gives the best predictive probability of the next target word is the same word that a human would choose. Another related limitation with our system is that it assumes that the previous system output is the same as the correct translation, and so the best next word to be translated by the system is the same as that of the reference translation.

As our approach is intended to reduce human effort, we look forward to conducting human subject experiments in future work, to see whether the gains we witnessed in simulation carry over to real-world conditions. One interesting direction that our method could provide is investigating whether the behaviors of humans interacting with such a system may be the same as those when interacting with other humans, and if not, to test in which ways human actions might be similar and how they may diverge from expected behavior. Another extension to this work would be incorporating the attention supervision into the main model. Currently, if asked to translate the same sentence twice, the current framework would ask for the same attention help twice, which seems inherently wasteful. Ideally, after getting the supervision, it would be able to incorporate it into the model to reduce redundant queries.

8 Conclusion

We have demonstrated an approach to interactive machine translation that aims to limit the amount of effort required by human translators while maintaining translation quality. We hope that our method inspires further research into this area.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate.

Britz, D., Goldie, A., Luong, M.-T., and Le, Q. (2017). Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

Foster, G. and Lapalme, G. (2002). *Text prediction for translators*. Université de Montréal.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Green, S., Wang, S. I., Chuang, J., Heer, J., Schuster, S., and Manning, C. D. (2014). Human effort and machine learnability in computer aided translation. In *EMNLP*, pages 1225–1236.

Hokamp, C. and Liu, Q. (2017). Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*.

Knowles, R. and Koehn, P. (2016). Neural interactive translation prediction. *AMTA 2016, Vol.*, page 107.

Koehn, P. (2009). A process study of computer-aided translation. *Machine Translation*, 23(4):241–263.

Langlais, P., Foster, G., and Lapalme, G. (2000). Transtype: a computer-aided translation typing system. In *Proceedings of the 2000 NAACL-ANLP Workshop on Embedded machine translation systems-Volume 5*, pages 46–51. Association for Computational Linguistics.

Mi, H., Wang, Z., and Ittycheriah, A. (2016). Supervised Attentions for Neural Machine Translation.

Ortiz-Martínez, D., García-Varea, I., and Casacuberta, F. (2010). Online learning for interactive statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 546–554. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, Montral, Canada.

Werling, K., Chaganty, A. T., Liang, P. S., and Manning, C. D. (2015). On-the-Job Learning with Bayesian Decision Theory. In *Advances in Neural Information Processing Systems*, pages 3465–3473, Montral, Canada.

Wuebker, J., Green, S., DeNero, J., Hasan, S., and Luong, M.-T. (2016). Models and inference for prefix-constrained machine translation. *54th ACL*, 1:66–75.

A Comparative Quality Evaluation of PBSMT and NMT using Professional Translators

Sheila Castilho[1] sheila.castilho@adaptcentre.ie
Joss Moorkens[1] joss.moorkens@adaptcentre.ie
Federico Gaspari[1] federico.gaspari@adaptcentre.ie
Rico Sennrich[2] rico.sennrich@ed.ac.uk
Vilelmini Sosoni[3] vilelmini@hotmail.com
Panayota Georgakopoulou[4] yota.georgakopoulou@bydeluxe.com
Pintu Lohar[1] pintu.lohar@adaptcentre.ie
Andy Way[1] andy.way@adaptcentre.ie
Antonio Valerio Miceli Barone[2] amiceli@inf.ed.ac.uk
Maria Gialama[4] maria.gialama@bydeluxe.com
[1]ADAPT Centre, Dublin City University, Dublin, Ireland
[2] The University of Edinburgh, Edinburgh, UK
[3] Ionian University, Corfu, Greece
[4] Deluxe Media, Athens, Greece

Abstract

This paper reports on a comparative evaluation of phrase-based statistical machine translation (PBSMT) and neural machine translation (NMT) for four language pairs, using the PET interface to compare educational domain output from both systems using a variety of metrics, including automatic evaluation as well as human rankings of adequacy and fluency, error-type markup, and post-editing (technical and temporal) effort, performed by professional translators. Our results show a preference for NMT in side-by-side ranking for all language pairs, texts, and segment lengths. In addition, perceived fluency is improved and annotated errors are fewer in the NMT output. Results are mixed for perceived adequacy and for errors of omission, addition, and mistranslation. Despite far fewer segments requiring post-editing, document-level post-editing performance was not found to have significantly improved in NMT compared to PBSMT. This evaluation was conducted as part of the TraMOOC project, which aims to create a replicable semi-automated methodology for high-quality machine translation of educational data.

1 Introduction

The industrial use of machine translation (MT) for production has become widespread since statistical machine translation (SMT) established itself as the dominant approach to translating texts automatically. Raw MT is now a viable solution for perishable content (Way, 2013) and post-editing of MT is offered by over 80% of language service providers surveyed by Lommel and DePalma (2016). In the years since the publication of Brown et al. (1993), an ecosystem of tools has grown around PBSMT, including scripts and tools for pre-processing and alignment, enabling incremental improvement in the quality of PBSMT output (Haddow et al., 2015).

More recently, the research community has become increasingly interested in the possibilities of neural machine translation (Bahdanau et al., 2014; Cho et al., 2014) (NMT), which

involves building a single neural network that maps aligned bilingual texts and, given input to translate, is trained to "maximize the probability of a correct translation" (Bahdanau et al., 2014) without external linguistic information. This interest is shared by many in the language service industry, where there is a need for improved MT quality and better quality estimation to "help reduce the frustrating aspects of post-editing" (Etchegoyhen et al., 2014). NMT results in the latest shared tasks have quickly matched or surpassed those of PBSMT systems, despite the many years of PBSMT development (Sennrich et al., 2016a; Bojar et al., 2016). Recent studies have reported an increase in quality when comparing NMT with PBSMT using either automatic metrics (Bahdanau et al., 2014; Jean et al., 2015), or small-scale human evaluations (Bentivogli et al., 2016; Wu et al., 2016). While these initial experiments with NMT have shown impressive results and promising potential, so far there have been a limited number of human evaluations of NMT output.

This paper reports the results of a quantitative and qualitative comparative evaluation of PBSMT and NMT carried out using automatic metrics and a small number of professional translators, considering the translation of educational texts in four language pairs, i.e. from English into German, Portuguese, Russian and Greek. It employs a variety of metrics, including side-by-side ranking, rating for accuracy and fluency, error annotation, and measurements of post-editing effort. This evaluation is part of the work for TraMOOC,[1] a European-funded project focused on the translation of MOOCs, which aims to create a replicable semi-automated methodology for high-quality MT of educational data. As such, the MT engines tested are built using generic and in-domain data from educational resources, as detailed in Section 3.1.1. The remainder of this paper is organized as follows: In Section 2 we review previous work comparing MT output using the statistical and neural approaches. We describe our MT systems and the experimental methodology in Section 3, and the results of human and automatic evaluations in Section 4. Finally, we draw the main conclusions of the study and outline promising avenues for future work in Section 5.

2 Previous Work Comparing PBSMT and NMT

A number of papers have been published recently which compare specific aspects of PBSMT and NMT. Bentivogli et al. (2016) asked five professional translators to carry out light post-editing on 600 segments of English TED talks data translated into German. These comprised 120 segments each from one NMT and four PBSMT systems. Using HTER (Snover et al., 2006) to estimate the fewest possible edits from pre- to post-edit, they found that technical post-editing effort (in terms of the number of edits) when using NMT was reduced on average by 26% when compared with the best-performing PBSMT system. NMT output showed substantially fewer word order errors, notably with regard to verb placement (which is particularly difficult when translating into German), and fewer lexical and morphological errors. Bentivogli et al. (2016) concluded that NMT has "significantly pushed ahead the state of the art", especially for morphologically rich languages and language pairs that are likely to require substantial word reordering.

Wu et al. (2016) used BLEU (Papineni et al., 2002) scores and human ranking of 500 Wikipedia segments that had been machine-translated from English into Spanish, French, Simplified Chinese, and vice-versa. Results from this paper again show that the NMT system strongly outperforms other approaches and improves translation quality for morphologically rich languages, with human evaluation ratings that were closer to human translation than PBSMT. The authors noted that some additional 'tweaks' would be required before NMT would be ready for real data, and Google NMT engines subsequently went live for the language pairs tested shortly after this paper was published (Schuster et al., 2016). Junczys-Dowmunt et al.

[1] http://tramooc.eu

(2016) also found BLEU score improvements in NMT when compared with PBSMT for as many as 30 language pairs.

Results of the 2016 Workshop on Statistical Machine Translation (WMT16) (Bojar et al., 2016) found that NMT systems were ranked above PBSMT and online systems for six of 12 language pairs for translation tasks. In addition, for the automatic post-editing task, neural end-to-end systems were found to represent a "significant step forward" over a basic statistical approach.

Toral and Sanchez-Cartagena (2017) compared NMT and PBSMT for nine language pairs (English to Czech, German, Romanian, Russian and vice-versa, plus English to Finnish), with engines trained for the news translation task at WMT16. BLEU scores were higher for NMT output than PBSMT output for all language pairs, except for Russian-English and Romanian-English. NMT and PBSMT outputs were found to be dissimilar, with a higher inter-system variability between NMT systems. NMT systems appear to perform more reordering than PB-SMT systems, resulting in more fluent translations (taking perplexity of MT outputs on neural language models as a proxy for fluency). Toral and Sanchez-Cartagena (2017) found that the tested NMT systems performed better than PBSMT for inflection and reordering errors in all language pairs. However, using the chrF1 automatic evaluation metric (Popović, 2015), which they argue is more suited to NMT, they found that PBSMT performed better than NMT for segments longer than 40 words.

Castilho et al. (2017) also reported on three comparative studies of PBSMT and NMT, discussing some of the preliminary results of the current study, highlighting some strengths and weaknesses of NMT, and the danger of hyperbole in discussions of the potential of NMT. Against this background, this paper attempts to shed more light on the emerging picture of the comparison between PBSMT and NMT.

3 Experiments

We built and evaluated PBSMT and NMT systems for four translation directions: English to German, Greek, Portuguese, and Russian. Evaluation was performed with automatic metrics, as well as with professional translators, who performed side-by-side ranking, adequacy and fluency rating, post-editing and error annotation based on a predefined taxonomy.

3.1 MT Systems

3.1.1 Training Data

The MT engines used in the TraMOOC project are trained on large amounts of data from various sources: the training data from the WMT shared translation tasks[2] and OPUS (Tiedemann, 2012) as mixed domain, and as in-domain training data we use TED from WIT3 (Cettolo et al., 2012); QCRI Educational Domain Corpus (QED) (Abdelali et al., 2014); a corpus of Coursera MOOCs; and our own collection of educational data. The amount of training data used is shown in Table 1.

Lang.	DE	EL	PT	RU
mixed domain	23.78	30.73	31.97	21.30
In-domain	0.27	0.14	0.58	2.31

Table 1: Training data size the EN→* translation directions (number of sentence pairs, in millions).

3.1.2 Phrase-based SMT

The PBSMT used is Moses (Koehn et al., 2007), MGIZA (Gao and Vogel, 2008) is used to train word alignments, and KenLM (Heafield, 2011) is used for LM training and scoring.

[2] http://www.statmt.org/wmt16/

The MT model is a linear combination of various features, including standard Moses features such as phrase translation probabilities, phrase and word penalty, and 5-gram LM with modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1998), as well as the following advanced features: a hierarchical lexicalized reordering model (Galley and Manning, 2008); a 5-gram operation sequence model (Durrani et al., 2013); sparse features indicating phrase pair frequency, phrase length, and sparse lexical features; and, for English-Russian, we employ a transliteration model for unknown words (Durrani et al., 2014). Feature weights are optimized to maximize BLEU with batch MIRA (Cherry and Foster, 2012) on an in-domain tuning set that has been extracted (and held out) from the in-domain training data.

Adaptation to the MOOC domain is performed via three mechanisms: sparse domain indicator features in the phrase table; linear interpolation of LMs with perplexity optimization on the in-domain tuning set; and learning of feature weights on the in-domain tuning set.

3.1.3 Neural MT

The NMT systems are attentional encoder-decoder networks (Bahdanau et al., 2014), which we trained with Nematus (Sennrich et al., 2017). We generally follow the settings used by Sennrich et al. (2016a). We use word embeddings of size 500, and hidden layers of size 1024, minibatches of size 80, and a maximum sentence length of 50. We train the models with Adadelta (Zeiler, 2012). The model is regularly validated via BLEU on a validation set, and we perform early stopping for single models. Decoding is performed with beam search with a beam size of 12.

To enable open-vocabulary translation, words are segmented via byte-pair encoding (BPE) (Sennrich et al., 2016c). For Portuguese, German, and Russian, the source and target sides of the training set for learning BPE are combined to increase consistency in the segmentation of the source and target text. For each language pair, we learn 89,500 merge operations.

For domain adaptation, we first train a model on all available training data, then fine-tune the model by continued training on in-domain training data (Luong and Manning, 2015; Sennrich et al., 2016b). Training is continued from the model that is trained on mixed-domain data, with dropout and early stopping. The models are an ensemble of 4 neural networks with the same architecture. We obtain the ensemble components by selecting the last 4 check-points of the mixed-domain training run, and continuing training each on in-domain data.

3.2 The MOOCs Domain

As this evaluation was intended to identify the best-performing MT system for the TraMOOC project, which focuses on high-quality MT for MOOCs, test sets were extracted from real MOOC data. These data included explanatory texts, subtitles from video lectures, user-generated content (UGC) from student forums or the comment sections of e-learning resources. One of the test sets was UGC from a business development course and the other three were transcribed subtitles from medical, physics, and social science courses. The UGC data was often poorly formulated and contained frequent grammatical errors. The other texts presented more standard grammar and syntax, but contained specialized terminology and, in the case of the physics text, non-contextual variables and formulae.

3.3 Materials, Evaluators, and Methods

For the purposes of this study, four English-language datasets consisting of 250 segments each (1K source sentences in total) were translated into German, Greek, Portuguese, and Russian using our PBSMT and NMT engines. The evaluation methods included two conditions: i) side-by-side ranking and ii) post-editing, assessment of adequacy and fluency, and error annotation. Both conditions were assessed by professional translators. More specifically, the ranking tasks consisted of only a subset (100 source segments) with their translations from PBSMT and NMT which were randomized and were carried out by 3 experienced professional translators (4 of

them in the case of Greek).The ranking was performed using Google forms.

For the second condition (ii), all the datasets (1K source sentences) were translated and the MT output (from both NMT and PBSMT) was mixed in each dataset, and the tasks were assigned in random order to the translators. The segments were presented sequentially, so as to maintain as much context as possible. These tasks were carried out by 3 experienced professional translators (2 in the case of English-German) using PET (Post-Editing Tool) (Aziz et al., 2012) over a two-week period. Participants were sent the PET manual and given PET installation instructions, a short description of the overall TraMOOC project and of the specific tasks, and requested to (in the following order) i) post-edit the MT output to achieve publishable quality in the final revised text, ii) rate fluency and adequacy (defined as the extent to which a target segment is correct in the target language and reflects the meaning of the source segment) on a four-point Likert scale for each segment, and iii) perform error annotation using a simple taxonomy (more details are provided in Section 3.5). This set-up had the advantage that measurements of two of Krings' (2001) categories of post-editing effort could be drawn directly from the PET logs, namely temporal effort (time spent post-editing) and technical effort (edit count).

3.4 Automatic Evaluation

The BLEU, chrF3 and METEOR (Banerjee and Lavie, 2005) automatic evaluation metrics are used in this study, with the caveat that two post-edits are used as references for each segment. It should be noted that Popović et al. (2016) suggest that the use of a single post-edited reference from the MT system under evaluation will tend to introduce bias. In addition, the HTER metric (Snover et al., 2006) was used to estimate the fewest possible edits between pre- and post-edited segments.

3.5 Human Evaluation

Ranking: The professional translators were asked to tick a box containing their preferred translation of an English source sentence for the side-by-side ranking task. PBSMT and NMT output was mixed and presented to participants using Google Forms. Two to three segments, where PBSMT and NMT output happened to be identical, were excised for each language pair, as the judges did not have the option to indicate a tie. The remaining tasks were carried out within the PET interface.

Adequacy and fluency rating: The judges were asked to rate adequacy in response to the question 'How much of the meaning expressed in the source fragment appears in the translation fragment?'. To avoid centrality bias, a Likert scale of one to four was used, where one was 'none of it' and four was 'all of it'. Similarly, fluency was rated on a one to four scale, where one was 'no fluency' and four was 'native'. Our expectation was that NMT would be rated positively for fluency, with possible degradation for adequacy, especially for longer segments (Cho et al., 2014; Neubig et al., 2015).

Post-editing and error annotation: Participants were asked to post-edit the MT segments to publishable quality, and then to highlight issues found in the MT output based on a simple error taxonomy comprising inflectional morphology, word order, omission, addition, and mistranslation. Again, our expectation was that there would be fewer morphology and word order errors with NMT, especially for short segments.

4 Results and Discussion

4.1 Automatic Evaluation

The automatic metric results using BLEU, METEOR, chrF3 and HTER are shown in Table 2. In particular, the decrease in word order errors in NMT output (as may be seen in Section 4.3)

shows an improvement in BLEU and METEOR scores, especially for some language pairs.

Table 2 shows that BLEU, METEOR and chrF3 scores considerably increase for German, Greek and Russian with NMT when compared to the PBSMT scores. These results were statistically significant in a one-way ANOVA pairwise comparison ($p<.05$) (marked with †). For Portuguese, moderate improvements can be observed, but no statistically significant differences were found.

Regarding the amount of PE that was required, the HTER scores show that more PE was performed when using the output from the PBSMT system for German, Greek and Russian. However, no statistically significant differences for HTER scores were found. The scores for chrF3 also show good improvement for NMT over PBSMT for German and Russian, but very similar results for Greek and Portuguese.

Lang.	System	BLEU	METEOR	chrF3	HTER
DE	PBSMT	41.5	33.6	0.66	49.0
	NMT	61.2 †	42.7 †	0.76	32.2
EL	PBSMT	47.0	35.8	0.65	45.1
	NMT	56.6 †	40.1 †	0.69	38.0
PT	PBSMT	57.0	41.6	0.76	33.4
	NMT	59.9	43.4	0.77	31.6
RU	PBSMT	41.9	33.7	0.67	44.6
	NMT	57.3 †	40.65 †	0.73	33.9

Table 2: Automatic Evaluation Results

4.2 Human Evaluation

Fluency and Adequacy: NMT was rated as more fluent than PBSMT for all language pairs. Table 3 shows the mean ratings for Fluency and Adequacy of the target languages for both PBSMT and NMT systems. Although no statistically significant differences were found, the percentage of scores assigned a 3-4 fluency value (Near Native or Native) for German is 68% for NMT as opposed to 54% for the PBSMT system, for Greek 75% and 65%, for Portuguese 80% and 74% , and for Russian 75% and 60%, respectively.

When looking at the percentage of scores assigned a 1-2 fluency value (No or Little Fluency) for each MT system's output, the NMT systems appear to have fewer problems when compared against the PBSMT systems for all the languages (German: 46% PBSMT vs. 32% NMT; Greek: 35% vs. 25%; Portuguese: 26% vs. 21%; and Russian: 40% vs. 25%).

A typical example of improved output for German NMT was the translation of the segment 'Would you send just 10 materials that are the most suitable.'

Lang.	System	Fluency	Adequacy
DE	PBSMT	2.60	2.85
	NMT	2.95	2.79
EL	PBSMT	2.86	3.44
	NMT	3.08	3.46
PT	PBSMT	3.15	3.73
	NMT	3.22	3.79
RU	PBSMT	2.70	2.98
	NMT	3.08	3.12

Table 3: Mean for Fluency and Adequacy

PBSMT: Würden Sie nur 10 Materialien, die am besten geeignet sind.
NMT: Schicken Sie einfach 10 Materialien, die am besten geeignet sind.

The German PBSMT output left out an infinitive verb at the end of the segment (literally 'Would you [polite form] just 10 materials that are the most suitable.'), while NMT produced a correct German translation, using the imperative verb form and retaining the correct register by using the 'Sie' politeness marker.

For Portuguese, one example of improved fluency is the translation of the segment 'I am

just making sure that I understand this correctly.'

PBSMT: Estou só para ter a certeza que entendi corretamente.
NMT: Eu estou apenas me certificando de que eu entendo isso corretamente.

The PBSMT system translates 'just' as 'só' (which in Portuguese can mean 'just' but as it is preceded by the verb 'estar', it implies the meaning 'alone'/'lonely'), conveying the misleading meaning of 'I'm alone to be sure if I understood correctly'. The NMT system translates 'making sure' as 'me certificando', which is accurate with the word 'just' translated as 'apenas' or 'só'.

One example for the Russian language is the translation of 'I liked your presentation a lot.'

PBSMT:Я любил свою презентацию много.
NMT: Мне очень понравилась ваша презентация.

While the NMT output is absolutely correct, the PBSMT system mistranslates the possessive pronoun 'your' as 'свою презентацию', which means 'my presentation'. It also translates 'liked' as 'любил', which means 'loved', and, finally, it also translates 'a lot' as 'много', which translates back as 'many' (quantifying adjective).

For Greek, NMT also shows improved fluency for the translation of 'What is the difference between a financial analyst and technical analyst and business analyst?'

PBSMT: Ποια είναι η διαφορά μεταξύ ένας οικονομικός αναλυτής και τεχνική αναλύτρια και οικονομικός αναλυτής·
NMT: Ποια είναι η διαφορά μεταξύ του οικονομικού αναλυτή και του τεχνικού αναλυτή και του επιχειρηματικού αναλυτή·

The NMT output is both semantically accurate and grammatically correct: the terms 'financial analyst', 'technical analyst' and 'business analyst' were rendered accurately in Greek, and, in addition, the nouns correctly appear in genitive form and the generic masculine is used. The PBSMT mistranslates the term 'business analyst' into 'οικονομικός αναλυτής' (i.e.'financial analyst'), and lacks fluency since the nouns are used in the nominative form and the gender of the noun 'technical analyst' appears in the feminine form rather than in the correct generic masculine form.

Regarding adequacy, however, results were overall less consistent (see Table 3) than those for fluency, with higher mean scores for German PBSMT. While NMT output received the highest mean ratings for all other language pairs, when considering 3-4 rankings (Most of It and All of It) as well as 1-2 rankings (None of It and Little of It), English-German PBSMT was ranked higher (73% against 66% for NMT), and English-Greek systems performed equally well (89% of the sentences assessed as 3-4 in terms of adequacy). For Portuguese and Russian, the NMT systems were ranked slightly higher when including 3-4 rankings, with PBSMT scoring 95% against 97% of NMT output in Portuguese, and for Russian the scores were 73% for PBSMT against 78% for NMT. These results are also replicated when the distinction between short and long sentences is made.

One example of adequacy for German where both MT systems committed errors can be seen in:

EN: We begin our exploration today by looking at a particular ad that appeared in on

American magazines in recent years.

PBSMT: Heute beginnen wir unsere Erforschung von einem bestimmten Ad anschaue, die auf amerikanischen Zeitschriften erschienen in den letzten Jahren.

NMT: Wir beginnen unsere Forschung heute mit einer bestimmten Werbung, die in den letzten Jahren in amerikanischen Zeitschriften veröffentlicht wurde.

The NMT output uses the noun 'Forschung', meaning 'research', rather than the correct 'Erforschung' as chosen by the PBSMT system. As a result, the participants rated this segment poorly for adequacy, and actually substituted the word 'Untersuchung' for 'exploration'. While the PBSMT system chose the correct noun, there were other word order and lexical errors that rendered the translation inadequate.

The following is an example of adequacy not being so consistent in translation into Portuguese, but NMT system still performing better:

EN: What we're going to need to do is, we're going to find the initial stretch, excuse me, the final stretch of the spring, the initial stretch of the spring, and subtract the squares.

PBSMT: O que vamos precisar fazer é, vamos encontrar o troço inicial, desculpe-me, o último troço da Primavera, o troço inicial da Primavera, e subtrair os quadrados.

NMT: O que vamos precisar fazer é, vamos encontrar o limite inicial, desculpe-me, o alongamento final da mola, o alongamento inicial da mola, e subtrair os quadrados.

PBSMT mistranslates the two main words of the sentence: 'stretch' (translates into 'stuff') and 'spring' (as the spring season, 'primavera'), thus making the translation unintelligible. NMT translates the term 'stretch' into two different ways ('limite' and 'alongamento'), but the sentence is still adequate and understandable.

For Russian, both MT systems also return errors for adequacy:

EN: We'll be drawing heavily on the field of art history and how interpretation works in that field.

PBSMT: Мы будем рисовать на области истории искусства и как интерпретации работает в этой области.

NMT: Мы будем активно рисоваться на области художественной истории и то, как интерпретация работает в этом поле.

Both systems translate the word 'drawing' as 'draw a picture'. PBSMT, however, retrieves a better translation for the remainder of the sentence, keeping 'история искусства' as a fixed expression, while 'художественной истории' – chosen by the NMT system – is not natural and the meaning is not clear. The translation of the word 'field' is also better in the PBSMT output: 'область' is 'field' in the sense of area (of research/interest), while NMT translates as 'поле', i.e. a farm field or mathematical concept.

Finally for Greek, the NMT system seems to handle adequacy a bit better:

EN: So, what if a resident or student wants to opt out of doing abortions?

PBSMT: Οπότε, τι γίνεται αν ένας κάτοικος ή μαθητής θέλει να εξαιρεθούν από το να κάνει εκτρώσεις·

NMT: Οπότε, τι γίνεται αν ένας κάτοικος ή φοιτητής θέλει να επιλέξει να κάνει εκτρώσεις·

The PBSMT translation has problems both at the level of fluency and at the level of adequacy,

while the NMT translation has problems only at the level of adequacy. In both the PBSMT and the NMT translations the term 'resident' - which in this context refers to the North American concept of 'a medical graduate engaged in specialised practice under supervision in a hospital'- is translated as 'κάτοικος', that is, a person who lives somewhere permanently or on a long-term basis. The PBSMT translates the word 'student' as 'μαθητής', which refers to a pupil, when in fact it should be translated as 'φοιτητής' (university student). The PBSMT output also suffers at the level of fluency due to the lack of subject to verb correspondence. In the NMT output, apart from the mistranslation of the term 'resident', there is one major mistranslation involving the phrasal verb 'opt out', as the NMT system translates it as 'opt', thus distorting completely the meaning of the source sentence.

Polysemous terms appear to pose the main problem to the NMT system for Greek and Russian languages, as it appears unable to discern semantic differences and choose the equivalent which bears the same meaning as the ST one in the translation. This can pose significant problems during the PE process, as translators may be misled by the inaccurate NMT rendering, and end up transferring the erroneous term in the final translation. For instance, for the translation of 'This is a magazine and a campaign called Got Milk where several famous figures appeared and they always asked the question, got milk?', the term 'figure' is translated into Greek by the PBSMT as 'προσωπικότητα', while it is translated erroneously by the NMT system as 'φιγούρα', which is semantically wrong. Another example of polysemous term appears in the Russian translation of 'Is it free?', where NMT translated as 'Свободно ли?', meaning 'unoccupied' ('is this seat/place free?'), while the PBSMT output includes a more frequent lexical item, 'Это бесплатно?', which relates to price ('free of charge'). For German and Portuguese, however, the polysemous terms are either not handled well by neither systems, or the NMT system provides a better translation.

This small selection of examples demonstrates the types of errors prevalent in the respective MT systems for each language pair studied, with the NMT output generally found to be more fluent and comprehensible, although not without errors. The type and prevalence of these errors throughout the test sets are detailed in Section 4.3.

4.3 Error Annotation

Category	DE		EL		PT		RU	
	PBSMT	NMT	PBSMT	NMT	PBSMT	NMT	PBSMT	NMT
Inflectional Morphology	732	608	443	307	404	378	695	506
	43%	49%	35%	28%	37%	37%	42%	38%
Word Order	382	180	303	208	216	181	197	122
	23%	15%	24%	19%	20%	18%	12%	9%
Omission	126	84	48	57*	53	58*	194	163
	7%	7%	4%	5%	5%	6%	12%	12%
Addition	46	39	24	31*	61	44	183	151
	3%	3%	2%	3%	6%	4%	11%	11%
Mistranslation	401	323	459	483*	348	342	385	404*
	24%	26%	36%	44%	32%	34*%	23%	30%
Total number of issues	1687	1234	1277	1086	1082	1003	1654	1346
Total number of "No Issues"	61	**189**	90	**168**	197	**236**	101	**195**
	6%	**18.9**%	9%	**16.8**%	19.7%	**23.6**%	10%	**19.5**%

Table 4: Error annotation

Table 4 shows the results of the error annotation task for all target languages, the total count of the errors and the percentage of errors of each category.[3] The total number of issues

[3]The percentage of errors is the number of error per category divided by the total number of errors found.

is greater for PBSMT than NMT for all language pairs. Moreover, the number of segments left without error annotations (No issues) is greater for NMT across all language pairs (in bold). NMT output was also found to contain fewer word order errors and fewer inflectional morphology errors in all the target languages. For English-Greek, the PBSMT output contained fewer errors of omission, addition, or mistranslation than NMT output (marked with an asterisk). For English-Portuguese, PBSMT showed fewer omissions and mistranslations, while English-Russian PBSMT contained fewer mistranslations (also marked with an asterisk).

Interestingly, the percentage of errors found in PBSMT and NMT seems to follow a pattern, with inflectional morphology, word order, and mistranslation being the most frequent problems found in both types of MT systems; with exception of the Russian language which presents a bit more mixed results for omission and addition. For German, inflectional morphology errors make up 49% of all the errors found in NMT output, a higher proportion than that found for PBSMT (where it accounts for 43% of the errors).

We therefore observe that the specific types of errors displayed by NMT and PBSMT output are to some extent dependent on the particular language pairs involved, and are clearly influenced by the specific morphosyntactic features of the target language. This, in turn, has implications for the post-editing effort involved in bringing the output to publishable quality, which will inevitably vary from one target language to another, also keeping the text type and the domain constant.

4.4 Ranking

For the ranking task, 400 English segments translated into Greek, and 300 segments translated into the other three target languages with NMT and PBSMT were compared side-by-side by professional translators who participated in the evaluation, using Google Forms. Participants preferred NMT output across all language pairs, with a particularly marked preference for English-German, as seen in Table 5. Inter-annotator agreement shows moderate agreement among the annotators (κ=0.60 for DE, κ=0.48 for EL, κ=0.40 for PT and κ=0.61 for RU).

Evaluation	**preference for**	
	PBSMT	**NMT**
EN-DE	61	239
(300)	20.3%	79.7%
EN-EL	174	226
(400)	43.5%	56.5%
EN-PT	115	185
(300)	38.3%	61.7%
EN-RU	110	190
(300)	36.7%	63.3%

Table 5: Ranking

This preference was consistent across all text types, with a 65% preference for NMT in the business analysis forum content, 54% preference for translations of a medical training transcript, 52% for translations of a physics transcript, and 55% for translations of an advertising transcript. Using distinctions from Pouget-Abadie et al. (2014), there was a 53% preference for NMT for short segments (20 tokens or fewer), and a 61% preference for NMT for long segments (over 20 tokens).

We believe that the text genres in which fluency is considered to be more important (i.e. business and marketing) have scored much better for NMT, as opposed to medicine and physics where a translator would tend to follow a more 'literal' translation, as it would typically be more important to translate all the words in the source, so as to ensure that the exact same meaning is preserved, sacrificing fluency if needed. We speculate that, for this reason, NMT may be a good fit for the subtitling domain in general, especially for material that is not particularly specialised.

4.5 Post-editing

Similarly to those segments left without error annotation, fewer NMT segments were considered by participants to require editing during the MT post-editing task. Table 6 shows the number of

segments changed and unchanged for all MT systems.

For German, the difference between the number of segments unchanged for NMT when compared with PBSMT output was very statistically significant in a one-way ANOVA pairwise comparison ($p<.05$, where M=.06, SE=.04) (marked with †). Table 7 shows the mean and standard deviation for temporal post-editing effort and Table 8 shows technical post-editing effort in the form of the average number of keystrokes per segment.

Lang.	System	Post-Edited	Unchanged
DE	PBSMT	940	60
	NMT	813	187†
EL	PBSMT	928	72
	NMT	863	137
PT	PBSMT	874	126
	NMT	844	156
RU	PBSMT	930	70
	NMT	848	152

Table 6: Unchanged Segments (out of 1000)

Average throughput or temporal effort was only marginally improved for German, Greek and Portuguese post-editing with NMT, as may be seen at the segment level in Table 7 and expressed in words per second in Table 9, while temporal effort for Russian was lower for PBSMT at the segment level.

Technical post-editing effort was reduced for NMT in all language pairs using measures of actual keystrokes (Table 8) or the minimum number of edits required to go from pre- to post-edited text (cf. the HTER scores in Table 2). Even though these results were not statistically significant, they suggest that those NMT segments that were edited required more cognitive effort than PBSMT segments. Feedback from the participants indicated that they found NMT errors more difficult to identify, whereas word order errors and disfluencies requiring revision were detected faster in PBSMT output.

Lang.	System	Mean	Std. Deviation
DE	PBSMT	74.8	21.12
	NMT	72.8	17.16
EL	PBSMT	77.7	1.85
	NMT	70.4	8.86
PT	PBSMT	57.7	14.23
	NMT	55.19	15.58
RU	PBSMT	104.6	3.62
	NMT	105.6	21.29

Table 7: Temporal Post-Editing Effort (secs/segment)

None of the participants reached the average rate of professional throughput, i.e. 0.39 words per second, found in Moorkens and O'Brien (2015) (Table 9): possibly with the exception of Portuguese, the translators remained quite far from this level of productivity, although it has to be stressed that this is heavily influenced by the type of text being translated as well as by the degree of expertise of the translators, not only with the subject matter at hand, but also, and crucially in the specific case reported here, with PE. This particular result may have also been affected by the unfamiliarity with the interface, the specialised nature of the texts and related research requirements, or perhaps the fact that the rating and annotation tasks carried out after post-editing disturbed the translators' momentum. Productivity is normally achieved with continuous work

Lang.	System	Mean	Std. Deviation
DE	PBSMT	5.8	1.84
	NMT	3.9	1.63
EL	PBSMT	13.9	0.16
	NMT	12.5	1.31
PT	PBSMT	3.8	1.68
	NMT	3.6	1.91
RU	PBSMT	7.5	4.99
	NMT	7.2	5.80

Table 8: Technical Post-Editing Effort (keystrokes/segment)

and translators/editors often report that their productivity peaks half-way into their day.

As for the distinction between long and short segments regarding the decision as to whether post-editing is required, the number of unchanged segments follows the same trend shown in Table 6, where fewer NMT segments were considered to require editing. In terms of words per second (see Table 10), the NMT system performs better with short sentences for German, Greek and Portuguese when compared to the PBSMT system, with the Portuguese language nearly reaching the average professional rate reported in Moorkens and O'Brien (2015).

Interestingly, the Russian output shows a slightly better WPS average for the PBSMT system for short sentences. Regarding long sentences, Greek and Russian show fewer WPS for NMT, but Portuguese and German show fewer WPS for the PBSMT system.

Similarly to the temporal effort results, the technical effort (keystrokes) results show that when distinguishing long and short sentences, German, Greek, and Portuguese present lower PE effort for NMT in short sentences, but the Russian output shows lower effort with PBSMT. For the long sentences, Greek and Russian show lower technical effort for NMT, whereas Portuguese and German show lower effort for the PBSMT system.

Lang.	PBSMT	NMT
DE	0.21	0.22
EL	0.22	0.24
PT	0.29	0.30
RU	0.14	0.14

Table 9: Words per Second (WPS)

	Lang.	PBSMT	NMT
Short	DE	0.21	0.26
(up to	EL	0.24	0.27
20 tokens)	PT	0.33	0.38
	RU	0.15	0.13*
Long	DE	0.21	0.20*
(greater	EL	0.20	0.22
than	PT	0.26	0.25*
20 tokens)	RU	0.13	0.14

Table 10: WPS: long vs short segments

5 Conclusions

This paper has presented the results of a large-scale comparative evaluation between NMT and PBSMT for four language pairs across several metrics, using complementary methods of human evaluation in addition to state-of-the-art automatic evaluation metrics, thus expanding the understanding of NMT's strengths and weaknesses compared to those of PBSMT. The study, that was conducted as part of the TraMOOC project, used translations of English educational domain data from real-life MOOCs into German, Greek, Portuguese, and Russian. For these language pairs and in this domain, we can conclude that fluency is improved and word order errors are fewer when using NMT, confirming the findings of other recent studies (see Section 2). Fewer segments require post-editing when using NMT, especially due to the lower number of morphological errors. There was, however, no clear improvement with regard to omission and mistranslation errors when moving from PBSMT to NMT. There was also no great decrease in PE effort, suggesting that NMT for production may not as yet offer more than an incremental improvement in temporal PE effort.

While overall NMT produced better results for our domain, expectations are high for NMT and financial pressures mean that the translation industry is eager for a leap forward in MT quality (Moorkens, 2017). At this juncture, however, the neural paradigm is not a panacea. Following on from this study, we intend to compare cognitive post-editing effort using average pause ratio (Lacruz et al., 2012) and to evaluate the effects of added in-domain data on NMT quality and domain specificity.

Acknowledgement The TraMOOC project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº644333. The ADAPT Centre for Digital Content Technology at Dublin City University is funded under the Science Foundation Ireland Research Centres Programme (Grant 13/RC/ 2106) and is co-funded under the European Regional Development Fund.

References

Abdelali, A., Guzman, F., Sajjad, H., and Vogel, S. (2014). The AMARA Corpus: Building Parallel Language Resources for the Educational Domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland.

Aziz, W., Castilho, S., and Specia, L. (2012). PET: a Tool for Post-editing and Assessing Machine Translation. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72.

Bentivogli, L., Bisazza, A., Cettolo, M., and Federico, M. (2016). Neural versus Phrase-Based Machine Translation Quality: a Case Study. *CoRR*, abs/1608.04631.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., and Zampieri, M. (2016). Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Castilho, S., Moorkens, J., Gaspari, F., Calixto, I., Tinsley, J., and Way, A. (2017). Is neural machine translation the new state of the art? *The Prague Bulletin of Mathematical Linguistics*, 108(1):109–120.

Cettolo, M., Girardi, C., and Federico, M. (2012). WIT3: Web Inventory of Transcribed and Translated Talks. In *Conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy.

Chen, S. F. and Goodman, J. (1998). An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, Cambridge, MA, USA.

Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 427–436, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *CoRR*, abs/1409.1259.

Durrani, N., Fraser, A., and Schmid, H. (2013). Model With Minimal Translation Units, But Decode With Phrases. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies NAACL*, pages 1–11, Atlanta, GA, USA.

Durrani, N., Sajjad, H., Hoang, H., and Koehn, P. (2014). Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014*, pages 148–153, Gothenburg, Sweden.

Etchegoyhen, T., Bywood, L., Fishel, M., Georgakopoulou, P., Jiang, J., Loenhout, G. V., Pozo, A. D., Maucec, M. S., Turner, A., and Volk, M. (2014). Machine Translation for Subtitling: A Large-Scale Evaluation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland.

Galley, M. and Manning, C. D. (2008). A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 848–856, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gao, Q. and Vogel, S. (2008). Parallel Implementations of Word Alignment Tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57, Columbus, OH, USA.

Haddow, B., Huck, M., Birch, A., Bogoychev, N., and Koehn, P. (2015). The Edinburgh/JHU Phrase-based Machine Translation Systems for WMT 2015. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 126–133, Lisbon, Portugal.

Heafield, K. (2011). KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Jean, S., Firat, O., Cho, K., Memisevic, R., and Bengio, Y. (2015). Montreal Neural Machine Translation Systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, Lisbon, Portugal.

Junczys-Dowmunt, M., Dwojak, T., and Hoang, H. (2016). Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions. In *Arxiv*.

Kneser, R. and Ney, H. (1995). Improved Backing-Off for M-gram Language Modeling. In *Proceedings of the Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184, Detroit, MI, USA.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the ACL-2007 Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Krings, H. P. (2001). *Repairing texts: empirical investigations of machine translation post-editing processes*. Kent State University Press, Kent, Ohio.

Lacruz, I., Shreve, G. M., and Angelone, E. (2012). Average Pause Ratio as an Indicator of Cognitive Effort in Post-Editing: A Case Study. In *AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP 2012)*, pages 21–30, San Diego, USA.

Lommel, A. R. and DePalma, D. A. (2016). Europe's Leading Role in Machine Translation: How Europe Is Driving the Shift to MT. Technical report, Common Sense Advisory, Boston, USA.

Luong, M.-T. and Manning, C. D. (2015). Stanford Neural Machine Translation Systems for Spoken Language Domains. In *Proceedings of the International Workshop on Spoken Language Translation 2015*, Da Nang, Vietnam.

Moorkens, J. (2017). Under pressure: translation in times of austerity. *Perspectives: Studies in Translation Theory and Practice*, 25(3).

Moorkens, J. and O'Brien, S. (2015). Post-editing evaluations: Trade-offs between novice and professional participants. In *Proceedings of European Association for Machine Translation (EAMT)*, pages 75–81, Antalya, Turkey.

Neubig, G., Morishita, M., and Nakamura, S. (2015). Neural Reranking Improves Subjective Quality of Machine Translation: NAIST at WAT2015. *CoRR*, abs/1510.05203.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Stroudsburg, PA, USA.

Popović, M. (2015). chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal.

Popović, M., Arcan, M., and Lommel, A. (2016). Potential and Limits of Using Post-edits as Reference Translations for MT Evaluation. *Baltic J. Modern Computing*, 4(2):218—-229.

Pouget-Abadie, J., Bahdanau, D., van Merrienboer, B., Cho, K., and Bengio, Y. (2014). Overcoming the Curse of Sentence Length for Neural Machine Translation using Automatic Segmentation. *CoRR*, abs/1409.1257.

Schuster, M., Johnson, M., and Thorat, N. (2016). Zero-Shot Translation with Google's Multilingual Neural Machine Translation System.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli Barone, A. V., Mokry, J., and Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation (WMT16)*, Berlin, Germany.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.

Sennrich, R., Haddow, B., and Birch, A. (2016c). Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200(6).

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey.

Toral, A. and Sanchez-Cartagena, V. M. (2017). A Multifaceted Evaluation of Neural versus Phrase-Based Machine Translation for 9 Language Directions. In *Conference of the European Chapter of the Association for Computational Linguistics*, EACL 2017. To Appear, Valencia, Spain. ACL.

Way, A. (2013). Traditional and Emerging Use-cases for Machine Translation. In *Translating and the Computer 35*, TC35, London, UK. ASLIB.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, abs/1609.08144.

Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701.

One-parameter models for sentence-level post-editing effort estimation

Mikel L. Forcada mlf@ua.es
Miquel Esplà-Gomis mespla@dlsi.ua.es
Felipe Sánchez-Martínez fsanchez@dlsi.ua.es
Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant,
E-03690 Sant Vicent del Raspeig, Spain.

Lucia Specia l.specia@sheffield.ac.uk
Department of Computer Science, the University of Sheffield,
Sheffield S1 4DP, United Kingdom

Abstract

Methods to predict the effort needed to post-edit a given machine translation (MT) output are seen as a promising direction to making MT more useful in the translation industry. Despite the wide variety of approaches that have been proposed, with increasing complexity as regards their number of features and parameters, the problem is far from solved. Focusing on post-editing time as effort indicator, this paper takes a step back and analyses the performance of very simple, easy to interpret one-parameter estimators that are based on general properties of the data: (a) a weighted average of measured post-editing times in a training set, where weights are an exponential function of edit distances between the new segment and those in training data; (b) post-editing time as a linear function of the length of the segment; and (c) source and target statistical language models. These simple estimators outperform strong baselines and are surprisingly competitive compared to more complex estimators, which have many more parameters and combine rich features. These results suggest that before blindly attempting sophisticated machine learning approaches to build post-editing effort predictors, one should first consider simple, intuitive and interpretable models, and only then incrementally improve them by adding new features and gradually increasing their complexity. In a preliminary analysis, simple linear combinations of estimators of types (b) and (c) do not seem to be able to improve the performance of the single best estimator, which suggests that more complex, non-linear models could indeed be beneficial when multiple indicators are used.

1 Introduction

Over the last decade, the interest of the industry in machine translation (MT) has grown, mainly as a consequence of high demand and improvements in translation quality. Modern MT systems have proven to lead to productivity gains (Plitt and Masselot, 2010; Guerberof Arenas, 2009) when used to generate draft translations that are then post-edited (corrected) before publishing (Krings and Koby, 2001; O'Brien and Simard, 2014). However, not all the translations produced by MT systems are worth post-editing. In some cases, it would be faster to translate them from scratch. As a result, a strong focus has been put into developing methods for estimating the quality of machine-translated sentences (Blatz et al., 2004; Specia et al., 2009) to identify those translations that may harm productivity if provided to post-editors. Several methods are

proposed every year and compared in the framework of the WMT series of Workshops on Machine Translation.[1]

Most approaches to MT quality estimation (QE) work at the sentence level, although there are also approaches that try to estimate the quality at the word or document levels. Sentence-level QE models predict translation quality in terms of post-editing (PE) time, number of edits needed, and other related metrics (Specia, 2011; Bojar et al., 2014). This paper focuses on sentence-level MT QE and measures quality in terms of PE time. This setting has the important advantage that the time predicted for each machine-translated sentence can be directly used to budget a translation job.

As will be discussed below, existing PE time estimators use many parameters and combine rich features extracted from source sentences and their raw MT output, often with the help of one or more pseudo-references obtained using additional MT systems. They are, however, still far from producing human-like predictions (with Pearson correlations between predicted and human effort metrics plateauing around 0.65, (Bojar et al., 2013, 2014)). To try to understand the problem better, we explore the use of three types of very simple, one-parameter, black-box PE time estimators: (a) a weighted average of PE times in the training set, where weights are an exponential function of edit distances computed between the current sentence (source or raw MT) and training sentences (source or raw MT), so that the contribution of nearest examples is more important; (b) a simple model that learns a unit PE time, either per character or per word, and multiplies it by the length of the current sentence (source or raw MT); and (c) logarithmic probabilities obtained by applying a statistical language model of the source or the target language respectively to the source or raw MT.[2]

The results show that some of these very simple models outperform not only rather strong baselines, but also some complex, multi-parameter estimators participating in the WMT13 (Bojar et al., 2013) and WMT14 (Bojar et al., 2014) PE time estimation contests. Results can be taken as an indication that one should take a step back and first analyse simple models with intuitive interpretations, to only then carefully and gradually increase their complexity, before blindly attempting sophisticated machine learning approaches. In a preliminary analysis, simple linear combinations of estimators of types (b) and (c) above does not seem to be able to improve the performance of the single best estimator, which may be taken as an indication that more complex, non-linear models should be considered when multiple indicators are used.

2 Settings and models

2.1 Corpora

We have conducted experiments using the data sets for English-to-Spanish (en→es) translation, which are publicly available as part of the quality estimation shared Task 1.3 of WMT13[3] (Bojar et al., 2013) and WMT14[4] (Bojar et al., 2014); Table 1 describes these data sets. For the experiments in this paper the corpora were pre-processed using the vanilla word tokenizer available in the Python NLTK package (Bird et al., 2009).

2.2 Notation and evaluation

The training data consists of a set of N triplets $\{(s_i, \mathrm{MT}(s_i), t_i)\}_{i=1}^{N}$ where s_i is a source sentence, $\mathrm{MT}(s_i)$ its raw MT output, and t_i the time taken to post-edit $\mathrm{MT}(s_i)$ into an adequate

[1] Last edition: `http://www.statmt.org/wmt17/quality-estimation-task.html`

[2] Language model features have already been proven to be the single best predictors in previous work, see e.g. (Felice and Specia, 2012; Shah et al., 2015).

[3] `http://www.statmt.org/wmt13/quality-estimation-task.html`

[4] `http://www.statmt.org/wmt14/quality-estimation-task.html`

	Translation direction	No. of segments Training	Test
WMT13	en→es	803	284
WMT14	en→es	650	208

Table 1: Translation direction and number of training and test instances for the corpora used in the experiments.

translation of s_i. The goal is to predict the PE time for a new set of M source sentences and their translations, $\{(s_j, \mathrm{MT}(s_j))\}_{j=1}^{M}$.

As in the WMT13 and WMT14 contests, performance will be measured over the test set as the *mean absolute error* (MAE) of the prediction $\hat{t}_j$, that is,

$$\mathrm{MAE} = \frac{1}{M}\sum_{j=1}^{M} |\hat{t}_j - t_j|.$$

In addition to this, Pearson's correlation r between the predicted and measured times will also be reported as a secondary comparison metric.

The best parameter for each model will be determined through minimization of the MAE over the training set, as will be explained in the next section.

2.3 Models

In what follows we describe the three one-parameter models we experimented with in order to predict PE time.

2.3.1 Weighted-average model (*Avg*)

This model estimates the PE time needed to turn $\mathrm{MT}(s_j)$ into an adequate translation of s_j as the weighted average

$$\mathrm{Avg}_u(\alpha, x_j) = \sum_{i=1}^{N} w(\alpha, x_i, x_j)\, t_i,$$

controlled by a single parameter α, whose weights $w(\alpha, x_i, x_j)$ depend on edit distances through

$$w(\alpha, x_i, x_j) = e^{-\alpha \mathrm{ED}_u(x_i, x_j)} / \sum_{i=1}^{N} e^{-\alpha \mathrm{ED}_u(x_i, x_j)},$$

where $\mathrm{ED}_u(x_i, x_j)$ is the edit distance between x_i and x_j, u is the unit used to compute it, either characters ($u = \mathrm{c}$) or words ($u = \mathrm{w}$), and x_i (resp. x_j) is either the source sentence s_i (resp. s_j) or its machine translation $\mathrm{MT}(s_i)$ (resp. $\mathrm{MT}(s_j)$). For positive values of α, the contribution $w(\alpha, x_i, x_j)$ of t_i diminishes with the distance between either the source sentences or between their raw machine translations. In particular:

- When $\alpha = 0$, $\mathrm{Avg}_u(0, x_j) = \frac{1}{N}\sum_{i=1}^{N} t_i$ for all j, that is, the arithmetic average of measured PE times; we will refer to this as the *naïve zero-parameter average*;
- When $\alpha \to +\infty$, the t_i corresponding to the minimum $\mathrm{ED}(x_i, x_j)$, that is, the nearest neighbour, is selected. In what follows, this predictor will be referred to as $\mathrm{NN}_u(x_j)$.

It is expected that a careful choice of α in $[0, +\infty)$ will give a better estimate by assigning a higher weight to closer examples. The weighted average effectively acts as a "soft nearest-neighbour" predictor.

To find the optimum value of α, the training corpus is randomly split in two sets: 80% of the samples are used to compute the edit distances and the remaining 20% are used as a development set.

The idea behind the weighted-average model bears some resemblance to the work by Béchara et al. (2016), where a *semantic textual similarity* (between the source sentences) is used to select a close example: instead of predicting time, Béchara et al. (2016) predict the BLEU score for sentences that do not have a reference translation available, using as reference that for the close example. Note the weighted-average model is clearly a *black-box* model, as it does have access to the inner workings of the MT system whose quality is being predicted. It is also an *example-based* model that computes a prediction for the current segment by looking up measured times for existing segments in a training set.

2.3.2 Models based on the PE time per segment length unit (*TLen*)

These very simple, one-parameter estimators predict the PE time t_j as

$$\mathrm{TLen}_u(a, x_j) = a\,\mathrm{len}_u(x_j),$$

where x_j is a source sentence s_j, or its machine translation $\mathrm{MT}(s_j)$, and $\mathrm{len}_u(x_j)$ is the length of x_j in characters ($u = \mathrm{c}$) or words ($u = \mathrm{w}$). Note that the coefficient a, which is obtained by directly minimizing the MAE over the whole training corpus, has an easy interpretation in seconds per character or seconds per word, respectively. Again, this is a *black-box* model, which, in addition, only looks at one property of the source or machine-translated segment: its length. When $x_j = s_j$, it simply predicts that PE time grows linearly with the source sentence. When $x_j = \mathrm{MT}(s_j)$, the estimate is similar if one assumes that target-segment length grows linearly with source-segment length. Note, however, that this predictor pays very little attention to the actual post-editability of the translation:

- *Any* MT output having the same length would have the same post-editing time, regardless of the actual target words.
- Truncated or abnormally short MT outputs would be consistently —and often incorrectly— estimated to be easier to post-edit.

These models are therefore expected to be very limited predictors of PE time.

2.3.3 Statistical language models

Source-language (SLM) and target-language models (TLM), trained on a subset of the WMT13 translation task data[5] (an interpolated combination of Europarl and News Commentary data) were used to compute the logarithm of the probability of s_j and $\mathrm{MT}(s_j)$, respectively. This is then multiplied by a coefficient a which is also optimized to minimize MAE on the whole training set. Language models are common indicators used in QE but also have important limitations as PE time predictors:

- A TLM basically measures the *fluency* of the translation (Specia et al., 2013, p. 80), and would estimate more fluent translations as easier to post-edit, regardless of their actual semantic relationship to the source sentence.
- A SLM would in contrast measure the *complexity* of the translation (Specia et al., 2013, p. 80), or, if the language model was trained on texts similar to those on which the MT system was trained, its *expectedness*. Nevertheless, its predictive power may be limited when applied to a system that was not trained on similar data (or to a rule-based system).

[5] `http://www.statmt.org/wmt13/translation-task.html`

It is however worth mentioning that language models are amongst the best performing features for sentence-level MT QE (Felice and Specia, 2012; Shah et al., 2015) and are therefore included in most models submitted to the WMT QE shared tasks.

3 Results and discussion

3.1 Performance of one-parameter predictors

Tables 2 and 3 summarize the results for the one-parameter models, placing them in the context of the results obtained by other WMT13 and WMT14 participants. The performance of the zero-parameter naïve average, that is, the one obtained using for all test segments the average time in the training set as a fixed estimate, and the four nearest-neighbour estimates $\mathrm{NN}_u(x_j)$ (see Section 2.3.1) are also provided for completeness. The main metric used in the discussion is MAE, the official metric in WMT13 and WMT14. Pearson correlations, also provided, roughly follow the same trend, and their comparison would lead to similar conclusions (but see Section 3.1.3 for a more detailed discussion).

3.1.1 WMT13 results

When ordering results by MAE, as in (Bojar et al., 2013), the one-parameter models (*Avg*, *TLen*, SLM and TLM) outperform at least 2 of the 14 participants, with *TLen* models actually outperforming 8 of them and the TLM outperforming 12 of them. The baseline system (Baseline bb17 SVR), using support vector regression and a well-known set of 17 black-box features (Specia et al., 2013) also outperforms 8 of the 14 participant models. It is worth mentioning that language models are also included as features in this baseline set; that is, the baseline system is a superset of the single-parameter models using LM features. Nevertheless, the TLM outperforms the baseline by a rather large margin. This result in particular may reveal problems not only present in the baseline but also in other participating submissions such as (a) additional features adding noise that the learning algorithm could not adequately handle, (b) the regression architecture used (for instance, support vector regression in the case of the baseline) not being adequate, (c) optimization not being good enough (for instance, due to an incorrect choice of hyperparameters or to incomplete convergence), or (d) over-fitting to a rather small training set. All these reasons are in principle possible and worth a closer examination. One of the participating systems is even outperformed by the naïve-average zero-parameter estimate, and two of them by one of the (also parameterless) nearest-neighbour estimates.

In general terms, computationally simpler (linear) *TLen* models perform better than the more complex (sum of exponentials containing edit distances) *Avg* models, while the outstanding performance of TLM and SLM may be explained by the fact that they were trained on the same data as the system whose quality was estimated — therefore, in this last case, the black-box assumption would not hold entirely.

3.1.2 WMT14 results

When ordering results by MAE, as in (Bojar et al., 2014), one-parameter models have a more modest performance in this dataset, beating only 3 out of the 10 submissions: one of them (FBK-UPV-UEDIN/NOWP), which uses hundreds of features obtained from the best 100,000 translations produced by a purposely-trained statistical MT system; another one, the baseline, a rather strong model (17 features), equivalent to the WMT13 baseline. Contrary to what happened for WMT13, character-level *Avg* models seem to perform slightly better than the *TLen* model and the SLM and TLM models; these language models were trained on the same data as for WMT13, whereas the MT systems evaluated in WMT14 were not. All zero-parameter models (naïve average, nearest-neighbour) rank below all participants.

We note that the performance of the official baseline system (Baseline bb17 SVR) is

particularly poor on this data set. The reason for that were the ranges used for the grid search to optimize the hyperparameters of the support vector machine model, which were different from those used in the WMT13 model. If the same ranges are used, the baseline reaches a MAE of 17.65, which would place it above all of the one-parameter models and above two of the participating systems. This issue shows further evidence that more complex models need to be carefully crafted, with special attention dedicated to their hyperparameters.

3.1.3 Analysis

How can length be such a reasonable estimator? In both datasets, length-based $\mathrm{TLen}_u(x_j)$ estimators show a rather competitive performance, in spite of the obvious limitations discussed in Section 2.3.2. This may be due to the fact that the output of a single MT system was post-edited and, therefore MT quality and, consequently, the post-editing effort across the segments produced by the MT systems is quite stable, effectively yielding a roughly constant per-word or per-character post-editing time and therefore making length a reasonable estimator in this case.[6] It would therefore be reasonable to expect performance to have been clearly worse if output from at least two MT systems with very different levels of quality had been post-edited.

Pearson correlations between predictions. In addition to the Pearson correlation with the test set, we have computed the Pearson correlation coefficient between the predictions of participating submissions —which are available at the WMT13[7] and WMT14[8] websites— and our best one-parameter *TLen*, *Avg*, and TLM models. In general terms, systems showing a good correlation with the one-parameter models happen to perform similarly, an indication that their predictions are very similar for test sentences. There are, however, interesting exceptions. An example of moderate correlation among predictors but similar performance is the SHEF FS submission to WMT13, which has correlation coefficient with $\mathrm{Avg}_\mathrm{c}(\alpha = 0.256, \mathrm{MT}(s_j))$ of 0.67 and an absolute difference in MAE of only 0.70. This may point at a certain complementarity between the two predictors, which seem to predict differently for many test sentences in spite of similar MAE performance. Another remarkable exception is the LIMSI elastic submission to WMT13, which correlates reasonably well with $\mathrm{TLen}_\mathrm{w}(a = 3.226, \mathrm{MT}(s_j))$ $(r = 0.74)$, $\mathrm{Avg}_\mathrm{c}(\alpha = 0.256, \mathrm{MT}(s_j))$ $(r = 0.75)$, and specially $\mathrm{TLM}(a = -1.421, \mathrm{MT}(s_j))$ $(r = 0.85)$ but performs considerably worse (the absolute differences in MAE are 18.6, 14.0, and 21.8 respectively). As we discuss in what follows, this could be an issue related to inadequate scaling of predictions.

Correlation and MAE leading to different system rankings: A scaling study. There are cases in which using the Pearson correlation obtained by participants does not lead to the same system ranking as that obtained by the official MAE-based ranking. One such a case is the LIMSI elastic submission, which has a Pearson correlation in the range of participants getting much lower MAE. Another interesting case is that of FBK-UPV-UEDIN/WP, FBK-UPV UEDIN/NOWP and RTM-DCU/RTM-RR in WMT14. Again, their Pearson correlation coefficients are clearly higher than those of other participants having similar MAE.

Discrepancies between MAE and correlation coefficients may easily be explained in terms of scaling; in fact, by simply scaling the outputs of all participating predictors one can obtain better MAE results, as shown in tables 2 and 3.

---

[6] The actual time per unit, both in the training set and the test set, indeed shows a rather peaked distribution density around the average values used by the length predictors.

[7] `http://www.statmt.org/wmt13/quality_estimation_data/QE_WMT13_submissions_task1.3_sentence.zip`

[8] `http://www.statmt.org/wmt14/quality_estimation_data/QE_WMT14_submissions_task1.3_sentence.zip`

System ID	MAE	r	Scaling	Scaled MAE	ΔMAE
FBK-UEDIN Extra	47.5	0.65	0.909	46.5	1.0
FBK-UEDIN Rand-SVR	47.9	0.66	1.062	47.6	0.3
$\mathrm{TLM}(a=-1.421, \mathrm{MT}(s_j))$	48.8	0.65			
CNGL SVR	49.2	0.67	1.164	47.6	1.6
CNGL SVRPLS	49.6	0.68	1.104	48.9	0.7
$\mathrm{SLM}(a=-1.249, s_j))$	49.7	0.64			
CMU slim	51.6	0.63	0.902	50.6	1.0
Baseline bb17 SVR	51.9	0.61	1.103	51.4	0.5
$\mathrm{TLen_w}(a=3.226, \mathrm{MT}(s_j))$	52.0	0.57			
$\mathrm{TLen_w}(a=3.468, s_j)$	52.3	0.59			
DFKI linear6	52.4	0.64	0.857	50.7	1.7
$\mathrm{TLen_c}(a=0.664, s_j)$	52.4	0.57			
$\mathrm{TLen_c}(a=0.601, \mathrm{MT}(s_j))$	52.5	0.57			
CMU full	53.6	0.58	1.006	53.6	0.0
DFKI pls8	53.6	0.59	0.874	52.1	1.5
TCD-DCU-CNGL SVM2	55.8	0.47	1.082	55.4	0.4
TCD-DCU-CNGL SVM1	55.9	0.48	1.083	55.5	0.4
SHEF FS	55.9	0.42	0.870	54.7	1.2
$\mathrm{Avg_c}(\alpha=0.256, \mathrm{MT}(s_j))$	56.6	0.53			
$\mathrm{Avg_c}(\alpha=0.386, s_j)$	57.2	0.56			
$\mathrm{Avg_w}(\alpha=1.079, \mathrm{MT}(s_j))$	61.1	0.52			
$\mathrm{Avg_w}(\alpha=0.612, s_j)$	61.7	0.59			
$\mathrm{NN}_c(s_j)$	62.5	0.41			
SHEF FS-AL	64.6	0.57	1.054	64.4	0.2
$\mathrm{NN}_c(\mathrm{MT}(s_j))$	67.8	0.35			
Naïve zero-parameter average	68.1	—			
$\mathrm{NN}_w(s_j)$	70.1	0.37			
LIMSI elastic	70.6	0.58	1.804	54.4	26.2
$\mathrm{NN}_w(\mathrm{MT}(s_j))$	71.3	0.30			

Table 2: Mean absolute error (MAE) and Pearson correlation coefficient (r) for one-parameter ($\mathrm{Avg}(\alpha, x_j)$, $\mathrm{TLen}(a, x_j)$, $\mathrm{SLM}(a, s_j)$ and $\mathrm{TLM}(a, \mathrm{MT}(s_j))$) and zero-parameter (naïve average, $\mathrm{NN}_u(x_j)$) quality estimators (all shaded) in the context of WMT13 submissions. For WMT13 participants, the results of *oracle scaling* (see text) are also given: scaling factor, new MAE and variation of MAE.

System ID	MAE	r	Scaling	Scaled MAE	ΔMAE
RTM-DCU/RTM-SVR	16.77	0.63	0.863	16.29	0.48
MULTILIZER/MLZ2	17.07	0.64	0.851	16.22	0.75
SHEFF-lite	17.13	0.61	0.949	17.05	0.08
MULTILIZER/MLZ1	17.31	0.65	0.835	16.43	0.88
SHEFF-lite/sparse	17.42	0.61	0.963	17.38	0.04
FBK-UPV-UEDIN/WP	17.48	0.66	0.812	15.76	1.72
RTM-DCU/RTM-RR	17.50	0.64	0.814	16.16	1.34
$\mathrm{Avg_c}(\alpha = 0.217, s_j)$	17.69	0.58			
$\mathrm{Avg_c}(\alpha = 0.202, \mathrm{MT}(s_j))$	17.94	0.57			
$\mathrm{TLM}(a = -0.538, \mathrm{MT}(s_j))$	18.38	0.57			
$\mathrm{TLen_c}(a = 0.281, \mathrm{MT}(s_j))$	18.55	0.58			
$\mathrm{SLM}(a = -0.521, s_j)$	18.59	0.55			
$\mathrm{TLen_w}(a = 1.519, MT(s_j))$	18.66	0.55			
FBK-UPV-UEDIN/NOWP	18.69	0.62	0.758	16.72	1.97
$\mathrm{Avg_w}(\alpha = 0.794, \mathrm{MT}(s_j))$	18.75	0.54			
$\mathrm{TLen_c}(a = 0.327, s_j)$	18.80	0.59			
$\mathrm{TLen_w}(a = 1.616, s_j)$	18.84	0.56			
$\mathrm{Avg_w}(\alpha = 0.61, s_j)$	18.86	0.56			
USHEFF	21.48	0.57	0.907	21.25	0.23
Baseline bb17 SVR	21.49	0.54	0.906	21.25	0.24
$\mathrm{NN}_c(s_j)$	21.53	0.37			
$\mathrm{NN}_w(\mathrm{MT}(s_j))$	21.80	0.36			
Naïve zero-parameter average	21.93	—			
$\mathrm{NN}_w(s_j)$	22.14	0.31			
$\mathrm{NN}_c(\mathrm{MT}(s_j))$	22.65	0.32			

Table 3: Mean absolute error (MAE) and Pearson correlation coefficient (r) for one-parameter ($\mathrm{Avg}(\alpha, x_j)$, $\mathrm{TLen}(a, x_j)$, $\mathrm{SLM}(a, s_j)$ and $\mathrm{TLM}(a, \mathrm{MT}(s_j))$) and zero-parameter (naïve average, $\mathrm{NN}_u(x_j)$) black-box quality estimators (all shaded) in the context of WMT14 submissions. For WMT14 participants, the results of *oracle scaling* (see text) are also given: scaling factor, new MAE and variation of MAE.

In the case of the LIMSI elastic submission to WMT13, the scaling factor of 1.804 leads to the best possible test-set MAE of 54.4, which is much better and closer to that of other systems having similar Pearson's coefficients. Note that this is an *oracle scaling*, since the gold-standard time measurements for the test set are used to obtain the best scaling factor; however, it is reasonable to expect that linear scaling on the training set would also have improved this predictor. For the remaining participants in WMT13, *oracle scaling* factors in the range $[0.857, 1.164]$ lead to small changes in MAE between 0.3 and 1.7 seconds, that is, around 0.6% to 3.4%. These changes would be expected to be even smaller or even negligible if scaling had been learned on the training set.

The scaling picture for WMT14 is also interesting (see Table 3). Oracle scaling factors in the range $[0.758, 0.949]$ lead to improvements in MAE in the range $[0.24\ s, 1.97\ s]$, which are sometimes as large as 12%. The improvements are particularly substantial for FBK-UPV-UEDIN/NOWP (–1.97 s, scaling 0.758), FBK-UPV-UEDIN/WP (–1.72 s, scaling 0.812) and RTM-DCU/RTM-RR (–1.34 s, scaling 0.814), which would explain the discrepancies between Pearson correlation and MAE mentioned above. It is reasonable to expect that a scaling factor obtained using the training set would have also made a difference in the test-set MAE in these three cases.

Approximating complex predictors with just one parameter: Finally, it is worth noting that some systems showing a good Pearson correlation with the models presented in this paper use very many features and parameters. In particular, the Pearson correlation coefficient of the RTM-DCU/RTM-SVR submission to WMT14 with $\mathrm{TLen_c}(a = 0.281, \mathrm{MT}(s_j))$ is 0.90 (the absolute difference in MAE is 1.78) and, while the latter has one feature and a single parameter, the former uses hundreds of features and several other sources of information. Oracle scaling of RTM-DCU/RTM-SVR slightly improves its test-set MAE to 16.23 s.

3.2 Performance of few-parameter predictors

In view of the surprisingly competitive results obtained with some of the single-parameter models presented here, one would immediately ask the following question: would performance improve further by using linear combinations of them?

We take the following six linear predicting features: the length-based $\mathrm{TLen_c}(s_j)$, $\mathrm{TLen_w}(s_j)$, $\mathrm{TLen_c}(\mathrm{MT}(s_j))$, and $\mathrm{TLen_c}(\mathrm{MT}(s_j))$, and the two statistical-language models SLM and TLM. For the study, we leave aside the weighted-average features as they are computationally more intensive to use and to train, do not have a linear form, and need a separate development set to be trained.

All $2^6-1 = 63$ possible subsets of these 6 features are studied.[9] We take linear combinations of each subset and use the multidimensional downhill simplex algorithm of Nelder and Mead (1965) as implemented in the Python library `scipy` to search the coefficients that minimize the training set MAE. For more than two parameters, the result of the minimization heavily depends on the starting point (this is expected in view of the strong collinearity, for instance, between length features). Therefore, and to ensure the best possible training set MAE, for each subset, 50 searches are performed with starting parameters randomly sampled from the zero-average, unit-variance normal distribution $\mathcal{N}(0, 1)$. The results are shown in Table 4.

As expected, the lowest training set MAE is found when all six features are used; however, the resulting test set MAE does not improve the results obtained with the best single-parameter predictor: 48.8 s for WMT13 (same as TLM alone) and 18.39 s for WMT14 (almost the same as TLM alone). Conversely, some combinations having worse training-set MAE get better test set MAE results, such as 48.22 s for a mixture of just $\mathrm{TLen_w}(s_j)$ and $\mathrm{TLM}(\mathrm{MT}(s_j))$ in WMT13,

[9] Exhaustive search in feature spaces is sometimes performed in QE, e.g. (Scarton et al., 2015).

Dataset	Features	Best combination	Train MAE	Test MAE
WMT13	1	TLM	41.3	48.8
	2	$\mathrm{TLen_w}(s_i) + \mathrm{TLM}$	41.0	48.2
	3	$\mathrm{TLen_w}(s_i) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{TLM}$	40.7	49.1
	4	$\mathrm{TLen_w}(s_i) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{SLM} + \mathrm{TLM}$	40.6	48.6
	5	$\mathrm{TLen_w}(s_i) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{TLen_w}(\mathrm{MT}(s_i)) + \mathrm{SLM} + \mathrm{TLM}$	40.5	48.8
	6	All 6	40.5	48.8
WMT14	1	SLM	15.92	18.59
	2	$\mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{SLM}$	15.60	18.44
	3	$\mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{SLM}$	15.57	18.51
	4	$\mathrm{TLen_c}(s_i) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{SLM}$	15.53	18.40
	5	$\mathrm{TLen_c}(s_i) + \mathrm{TLen_w}(s_i) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{TLen_c}(\mathrm{MT}(s_i)) + \mathrm{SLM}$	15.53	18.40
	6	All 6	15.53	18.39

Table 4: Post-editing time prediction using a small number of linear features: number of features, best combination, training-set MAE, and test-set MAE.

or 18.2 s for a mixture of $\mathrm{TLen_w}(s_j)$, $\mathrm{TLen_c}(\mathrm{MT}(s_j))$ and $\mathrm{TLM}(\mathrm{MT}(s_j))$ for WMT14. These results may be a possible indication of over-fitting or a limitation of a simple linear regressor.

3.3 Budgeting translation jobs

An interesting use of PE time predictors is *budgeting* a PE job, when post-editors are paid by the hour. Given a new translation job, an estimate of time to complete that job may easily be obtained by summing up the predicted PE time over all segments. This is a very practical application of QE.

Disregarding the actual hourly rate (a constant factor), a good estimate of the usefulness for budgeting may be given by studying the Pearson correlation between the total time predicted for a job by a certain estimator and the actual total time for that job.

To simulate that, we repeatedly and randomly extract PE jobs $\{(s_j, \mathrm{MT}(s_j), t_j)\}_{j=1}^{n}$ of $n = 100$ sentences from each of the test sets without replacement. Over each one of these sets, we compute the Pearson correlation between the predicted total time and the actual total measured time. The actual regression coefficients obtained vary with the number of random jobs, but their values for job sizes of 0.4, 0.8, 1.0, and 2.0 times the size of the test set and for a fixed number of 1000 jobs show consistent relative trends. The results for a number of jobs equal to the number of segments in the test set are shown in Table 5.

As can be seen, the Pearson correlation reported for the best single-parameter predictors is almost the same as that for the winning system in WMT13, and slightly worse in WMT14. This would suggest that, at least for these datasets, simple predictors could be used instead of very complex predictors having a large number of features and parameters with a very small loss in budgeting accuracy.

Dataset	Predictor	r
WMT13	FBK-UEDIN Extra (winner)	0.867
	$\mathrm{TLM}(\mathrm{MT}(s_j))$	0.856
	$\mathrm{SLM}(s_j)$	0.854
	$\mathrm{TLen_w}(\mathrm{MT}(s_j))$	0.856
	$\mathrm{Avg_c}(\mathrm{MT}(s_j))$	0.806
	Baseline	0.849
WMT14	RTM-DCU/RTM-SVR (winner)	0.860
	$\mathrm{Avg_c}(s_j)$	0.821
	$\mathrm{TLM}(\mathrm{MT}(s_j))$	0.828
	$\mathrm{TLen_c}(\mathrm{MT}(s_j))$	0.827
	$\mathrm{SLM}(s_j)$	0.819
	Baseline	0.730

Table 5: *Budgeting* Pearson correlation coefficients for selected PE time predictors, computed for a number of random jobs equal to the number of segments in the test set.

4 Concluding remarks

The results obtained by very simple, one-parameter MT QE models happen to be surprisingly competitive with those obtained by complex QE models using strong learning algorithms, tens, hundreds or thousands of features, and, sometimes, additional resources such as existing, custom-trained, or external MT systems. The findings in this study lead us to make the following recommendations for researchers in MT QE:

- First, look at what can be done with very simple models before *using a sledgehammer to crack nuts*, in order to get an idea of the performance one could obtain and hopefully improve. As some of the features used in the simple models proposed here are usually part of participants' complex models, the modest performance they obtain may be due to noise introduced by new features that could not be filtered out by the regressors (probably as a result of a non-optimal training process), to learning problems such as over-fitting to the training set, to non-optimal hyper-parameter choice, to incomplete convergence, or to the shortcomings of the regressors used (as revealed by the oracle scaling described in Section 3.1.3); the actual reasons are probably worth a closer analysis.
- Then, incrementally explore more complex models; linear combinations of a few carefully selected features do not seem to help much; therefore, one should probably consider simple non-linear models. The results of this analysis may be expected to shed some light on the problem.

Finally, a better understanding of the contribution of each feature to the QE models using them could open the door to using, in real-life QE scenarios, feasible and computationally simpler predictors.

Acknowledgements: Work supported by the Spanish government through the EFFORTUNE (TIN2015-69632-R) project and through grant PRX16/00043 for Mikel L. Forcada, and by the European Commission through the QT21 project (H2020 No. 645452).

References

Béchara, H., Parra-Escartín, C., Orăsan, C., and Specia, L. (2016). Semantic textual similarity in quality estimation. *Baltic J. Modern Computing*, 4(2):256–268.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media, Inc.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., and Ueffing, N. (2004). Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pages 315–321.

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the 8th Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., et al. (2014). Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the 9th Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, MD, USA.

Felice, M. and Specia, L. (2012). Linguistic features for quality estimation. In *Proceedings of the 7th Workshop on Statistical Machine Translation*, pages 96–103, Montreal, Quebec, Canada. Association for Computational Linguistics.

Guerberof Arenas, A. (2009). Productivity and quality in the post-editing of outputs from translation memories and machine translation. *The International Journal of Localisation*, 7(1):11–21.

Krings, H. P. and Koby, G. S. (2001). *Repairing texts: empirical investigations of machine translation post-editing processes*, volume 5. Kent State University Press.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.

O'Brien, S. and Simard, M. (2014). Introduction to special issue on post-editing. *Machine Translation*, 28(3-4):159–164.

Plitt, M. and Masselot, F. (2010). A productivity test of statistical machine translation post-editing in a typical localisation context. *The Prague Bulletin of Mathematical Linguistics*, (93):7–16.

Scarton, C., Tan, L., and Specia, L. (2015). USHEF and USAAR-USHEF participation in the WMT15 QE shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 336–341.

Shah, K., Cohn, T., and Specia, L. (2015). A Bayesian non-linear method for feature selection in machine translation quality estimation. *Machine Translation*, 29(2):101–125.

Specia, L. (2011). Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 73–80.

Specia, L., Shah, K., De Souza, J. G., and Cohn, T. (2013). QuEst - a translation quality estimation framework. In *Proceedings of the Conference of the Association of Computational Linguistics (Conference System Demonstrations)*, pages 79–84.

Specia, L., Turchi, M., Cancedda, N., Dymetman, M., and Cristianini, N. (2009). Estimating the Sentence-Level Quality of Machine Translation Systems. In *13th Annual Conference of the European Association for Machine Translation*, pages 28–37, Barcelona, Spain.

A Minimal Cognitive Model for Translating and Post-editing

Moritz Schaeffer moritzschaeffer@gmail.com
Gutenberg University, Mainz, Germany

Michael Carl m.gummiball@gmail.com
Renmin University of China, and Copenhagen Business School, Denmark

Abstract

This study investigates the coordination of reading (input) and writing (output) activities in from-scratch translation and post-editing. We segment logged eye movements and keylogging data into minimal units of reading and writing activity and model the process of post-editing and from-scratch translation as a Markov model. We show that the time translators and post-editors spend on source or target text reading predicts with a high degree of accuracy how likely it is that they engage in successive typing. We further show that the typing probability is also conditioned by the degree to which source and target text share semantic and syntactic properties. The minimal cognitive Markov model describes very basic factors which play a role in the processes occurring between input (reading) and output (writing) during translation.

1 Introduction

We build a cognitive model of the translation process (from-scratch translation and post-editing) which aims at predicting where translation problems occur. We ground the model in translation activity data that consists of keystrokes and gaze data that was captured during translation sessions. We decompose the translation process into minimal cycles of iterative reading and writing. We assume that the typing activities represent the solution to a translation problem that emerged during the preceding reading event. We show that the complexity (i.e. non-literality) of the produced translation as well as the duration and distribution of gaze activities on the source and target texts has an effect on the probability of a successive typing event.

Schaeffer et al. (2016); Hvelplund (2016); Carl et al. (2016); Läubli and Germann (2016) describe methods to decompose the stream of eye movements and keystrokes into sequences of minimal activity units. In this paper we relate the duration of activity units with properties of the translation product — the degree of translation literality — to predict the probability when post-editors and translators will type after reading either the source (henceforth ST) or the target text (henceforth TT).

Carl et al. (2016) show that a measure of *translation literality* has a great predictive power for behavioral observations in the translation process. According to this definition, a translation is literal if:

1. Word order is identical in the ST and TT

2. ST and TT items correspond one-to-one

3. Each ST word has only one possible translated form in a given context

A translations which completely fulfills all three criteria is an *absolutely literal translations*. A *literal* translation consists of the same number of ST and TT tokens where each TT token corresponds to exactly one ST token, and tokens in both texts are ordered in the same way. A change in word order or a situation in which one ST word is aligned to more than one TT word or vice versa weakens literality criteria 1 and 2 and makes a translation less literal. Criteria 1 and 2 thus measure the *syntactic similarity* of an ST and its translation. The third criterion describes the *semantic similarity* in both languages. If a word (or phrase) is consistently translated in the same way by different translators, we assume that the ST word and its translation also have large overlapping semantic properties. The more a source word (or phrase) can be rendered into different translations, the weaker is also the semantic overlap between the two languages (with respect to this word or phrase). In this paper we show that the degree of translation literality has an effect on the reading activities prior to translation typing.

In section 2 we introduce an operationalization of the literality metric as described above. We introduce a metric "HCross" which measures the entropy of word-order choices that are observed in alternative translations, and which is strongly predictive for reading time duration during the translation process. Section 3 presents the material of our empirical study. In section 4 we introduce translation units and translation states, as well as the topology of a minimal cognitive model for from-scratch translation and post-editing. We review similar work which used transition networks of activity units to model novice and expert translators. We review a proposal that defines different translation styles and map these onto sequences of translation states of our minimal cognitive model. In section 5 we analyze our data and develop a minimal model of translation and post-editing.

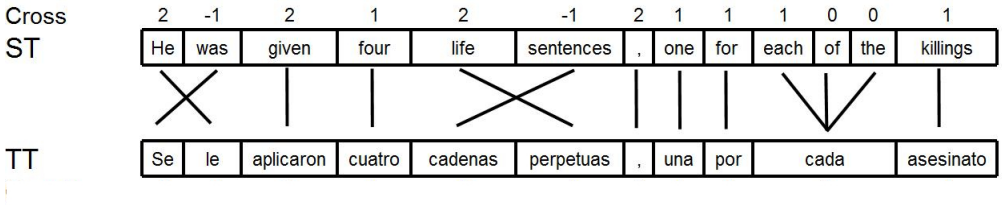


Figure 1: An English-Spanish alignment with Cross values

2 Operationalising Translation Literality

2.1 Word-order Distortion (Cross)

From a given translation and its word alignment relations we compute Cross values (see Figure 1). For any two successive source words s_{k-1} and s_k, we follow the alignment links to their translations ($s_{k-1} \rightarrow t_{k-1}$ and $s_k \rightarrow t_k$}) and compute the distance between the position of words t_k and t_{k-1} in the translation (i.e. $\text{position}(t_k) - \text{position}(t_{k-1})$) as the value for $\text{Cross}(s_k)$. We thus obtain a vector of relative alignment distortions for word positions in the ST and the TT, indicating the word order similarity of the two sentences. In the case of an (absolutely) literal translation, we say that each successive word aligns with the next one in the target language, which provides the Cross vectors with values 1.

For instance, the word [He] in Figure 1 occurs at position 1 on the English source side while its Spanish translation [le] occurs one word ahead at position 2 in the translation. [He] thus has a $\text{Cross}(s_1)$ value of 2 in that sentence. In order to generate the translation [aplicaron] for the English [given] we need to jump from the previous alignment [was-Se] two words to the

right, which produces a $\mathrm{Cross}(s_3)$ value of 2. In this way, Cross values are generated for each word position in the text, for the source and the target sides. If a word is aligned to more than one word (e.g. $s_k \rightarrow \{t_{k_1} \dots t_{k_n}\}$), $\mathrm{Cross}(s_k)$ is the signed value of the maximum absolute difference between the two translations, i.e. $\max(abs(\{t_{k_1} - t_{k-1}), \dots, abs(t_{k_n} - t_{k-1})\})$. In this way, t_{10} which has the alignment [cada → "each of the"] has an alignment distortion value $\mathrm{Cross}(t_{10}) = 3$.

2.2 Word Translation Entropy (HTra)

Carl et al. (2016) introduce word translation entropy as a measure to quantify observed translation choices. Entropy, H, represents the average amount of non-redundant information provided by each new item. It is computed based on the sum of the probability of the items and their information. The information of a probability p is defined as $I(p) = -\log_2(p)$ The entropy H is the expectation of that information as defined in equation (1):

$$H = \sum_{i=1}^{n} p_i I(p_i) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{1}$$

We adopt this notion to assess the entropy of word translation choices for a given ST word s_k into its n possible translations $t_{i\dots n}$ as shown in equation (2)

$$\mathrm{HTra}(s_k) = -\sum_{i=1}^{n} p(t_i|s_j) \times \log_2(p(t_i|s_j)) \tag{2}$$

The word translation entropy $\mathrm{HTra}(s_k)$ in equation 2 is computed for each source word s_k and in every segment. The translations $t_{i\dots n}$ are taken only from the aligned alternative translations of this segment. That is, the word translation probabilities $p(t_i|s_k)$, as computed according to equation (3), represent the ratio of the number of observed translations $s_k \rightarrow t_i$ separately for each source segment in which s_k occurs. Thus, while in language modeling, the entropy indicates how many possible continuations for a sentence exist at any time, we deploy the metric to assess how many different translations an ST word has in a given context.

$$p(t_i|s_k) = \frac{\mathrm{count}(s_k \rightarrow t_i)}{\mathrm{count}(s_k)} \tag{3}$$

We take it, that HTra reflects the semantic similarity between a source word and its translation(s): low HTra values indicate a high amount of agreement between translator choices, and thus a high degree of semantic similarity according to literality criterion 3 above.

2.3 Word-order Entropy (HCross)

The choices that a translator has to re-order translations of a source word s_k in the target language is captured by the metric HCross, as given in equation 4.

$$\mathrm{HCross}(s_k) = -\sum_{i=1}^{n} p(\mathrm{Cross}(s_k)) \times \log_2(p(\mathrm{Cross}(s_k)) \tag{4}$$

The probability for each relative translation word-order distortions $p(\mathrm{Cross}(s_k))$ for a source word s_k is computed as the ratio of the number of the distortions $\mathrm{Cross}(s_k)$ for alternative translations $s_k \rightarrow t_{1\dots n}$ divided by the total number of observed alternative translations $\mathrm{count}(s_k)$, similar to equation 3.

HTra and HCross values correlate to a high degree (*r*=.79, $p < .001$). That is, semantic and syntactic variation seem to correlate highly in translation. More variation in syntactic (i.e.

word-order) rendering of the translation seem to come along with more variation in lexical choices, and vice versa: Low cross-lingual semantic similarity (i.e. high HTra values) are correlated with high syntactic variation and complexity (i.e. high HCross values).

3 Experimental material

As a basis for our investigation in this paper we use the *multiLing* subset of the TPR-DB (Carl et al., 2016). The *multiLing* set consists of six short English source texts (together 849 words, 40 ST segments) and a large number of alternative translations into Danish (da), Spanish (es), German (de), Hindi (hi), Chinese (zh) and Japanese (jp) each by several translators. It contains currently more than 1500 text production sessions, for from-scratch translation (T), post-editing (P), monolingual editing (E), translation dictation (D) and text copying (C). However, in this study we only make use of from-scratch translation and post-editing, which amounts to approximately half the data, 124 hours productin time. For each text production session, keystroke and gaze data were collected and stored. A real-time gaze-to-word mapping tool (Carl, 2012) was used to map the gaze samples on the words, so that it is known which word was gazed at, at any time during the translation sessions. The tool also computes which keystroke contributes to the production (or modification) of which word. The STs and TTs were manually aligned using the YAWAT tool (Germann, 2008). Aligners were adviced to align each segments as compositional and complete as possible. The aligned data were further post-processed into a set of summary tables, which integrate and describe the data of the translation process and the translation product by means of currently more than 300 features (Carl et al., 2016).

	SText:#Seg		1:6	2:7	3:5	4:5	5:10	6:7	STtok	STseg	
	ST Token		160	154	146	110	139	139	848	40	
Task	Study	TL	Alt	Alt	Alt	Alt	Alt	Alt	TTtok	TTseg	Dur
P	BML12	es	10	12	10	12	8	12	10216	431	5.22
	ENJA15	ja	13	12	14	12	13	12	14447	519	16.81
	MS12	zh	3	5	3	3	3	2	2561	129	3.18
	NJ12	hi	7	12	8	10	12	11	9365	409	18.2
	SG12	de	8	7	7	8	7	8	6470	305	8.78
T	BML12	es	11	10	8	10	12	8	9938	411	10.34
	ENJA15	ja	12	13	12	13	13	13	14134	525	22.46
	KTHJ08	da	24	23	22	0	0	0	10667	523	7.7
	MS12	zh		3	3	3	3	3	1916	89	4.12
	NJ12	hi	7	7	5	7	6	6	5783	266	14.84
	SG12	de	6	8	8	8	7	8	6777	305	12.46
	Total		**101**	**112**	**100**	**86**	**84**	**83**	**92274**	**3912**	**124**

Table 1: Subset of the TPR-DB *multiLing* corpus with the post-editing (P) and from-scratch translation (T) data. The table shows for each of the six English source texts the number of segments and the number of words, as well as the total number of ST segments (STseg:40) and words (STtok:848). It also shows for each language the number of alternative translations (Alt) the total number of target text tokens (TTtok), segments (TTseg) and duration (Dur) per target language and for each of the translation modes.

Table 1 shows some figures of *multiLing* Corpus. The length in words for each of the six STs is given in the first row in Table 1 (ST1-6). For each of the six STs, the table indicates the number of participants (#Part), and for each of the six STs the number of alternative translations (Alt) and their total number in tokens (TokT). The total number of target words (TtokT) and

target sentences (Ttsg) is also provided, together with the total production duration in hours (Dur). The data is freely available. For more information on this dataset, please consult the CRITT website.[1]

4 Translation states

We extend the work of Schaeffer et al. (2016), who introduce Activity Units as a means to segment the stream of translation (and post-editing) activity into distinct units. Similar to Carl et al. (2016) they make a distinction between 6 different basic types of activities [2]:

- type 1: ST reading
- type 2: TT reading
- type 4: translation typing (no gaze data recorded)
- type 5: ST reading and typing (touch typing)
- type 6: TT reading and typing (translation monitoring)
- type 8: no gaze or typing activity recorded for more than 2.5 seconds

In this study we simplify the 6 types of activity units into four translation states. We collapse activity units 4, 5 and 6 into writing activities (W), irrespectively of whether reading activities are also recorded at the same time. This leaves us with the following four translation states:

	Post-editing						From-scratch translation					
	# OBS	%Dur	S_2	T_2	W_2	P_2	# OBS	%Dur	S_2	T_2	W_2	P_2
S_1	15695	26	0.00	**0.81**	0.16	0.02	17756	29	0.03	**0.52**	0.42	0.03
T_1	19275	**40**	**0.56**	0.01	0.41	0.03	17417	19	0.42	0.00	**0.54**	0.03
W_1	13092	27	0.35	**0.44**	0.14	0.07	26187	**44**	**0.36**	0.28	0.30	0.05
P_1	1723	8	0.19	0.28	**0.53**	0.00	2303	8	0.18	0.21	**0.60**	0.00
Total	49785	38.76 hours					63663	42.44 hours				

Table 2: Distribution of translation states in number of total observations (#OBS) and duration (%Dur), as well as a transition matrix for post-editing and from-scratch translation. The data represents translation states during the drafting phase of the data from Table 1

- S: ST reading (with no concurrent writing activity)
- T: TT reading (with no concurrent writing activity)
- W: Writing (with or without concurrent gaze activity on the source or target window)
- P: Pausing (no activity recorded for more than 2.5 seconds)

Each of the translation states (i.e activity units) can be described by a number of features (excluding P which has only a duration), including the number of keystrokes (deletions and insertions), the word(s) produced by the keystrokes, the number and duration of fixations, the fixation scanpath (i.e. sequence of fixations) within a state, including the number of different words fixated, their average distance etc. (cf. Schaeffer et al. (2016)).

[1] sites.google.com/site/centretranslationinnovation

[2] The Activity Unit of type 7, as suggested in Carl et al. (2016), which entails concurrent type 1, 2 and 4 behaviour is not assumed here. Instead the activities were split into the six types above.

4.1 State transitions in translation and post-editing

The data in Table 2 shows the distribution of translation states from the *mulitLing* data which were introduced in Table 1. The total dataset was segmented into 49,785 and 63,663 activity units for the post-editing and translation experiments respectively.

The data represented in Table 2 only accounts for the activities during the drafting phase. This amounts to 38.76 hours post-editing and 42.44 hours translating. The column #OBS shows the number of observations per translation state, while the %Dur column gives their percentage of the total production duration. In the post-editing mode, most activities (19,275 units) were observed in the TT reading (T_1) mode, as well with respect to the number of units and with respect to their duration. In the translation mode, the translators were 44% of the total time involved in writing activities.

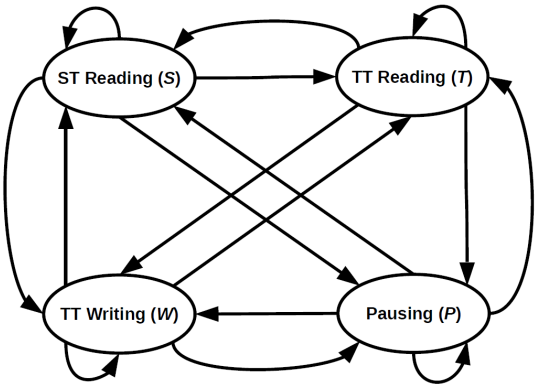


Figure 2: A fully connected translation process transition network with four states.

The columns S_2, T_2, W_2 and P_2 provide the likelihood of the next state to which post-editors or translators will switch.[3] For instance, if a post-editor is involved in an ST reading state (S_1), there is a high chance of 81% that next he or she will switch to TT reading (T_2). Once in the T_1 state, the highest probability (56%) is to switch back to ST reading (S_2). This is different in the translation mode, where the translator will most likely turn to writing (W_2) after a T_1 activity. Table 2 provides thus a transition table which can be represented in the form of a completely connected transition network as shown in Figure 2. Each state in the network in Figure 2 is connected to each other state in the network, and the transition from one state to the successor states are weighted by probabilities, such that the sum of all outgoing archs sums to 1.0. Two possible instantiations of the transition network are shown in Table 2, which produce slightly different behavior for post-editing and for from-scratch translation.

4.2 Novice and expert translators

Hvelplund (2016) reports that novice and expert translators exhibit different behavior with respect to the length and the sequencing of translation activities. His study is restricted to the English to Danish data collection which is gathered in the KTHJ08 study in Table 2. According to Hvelplund, experienced translators shift more often from ST reading (S_1) directly to writing (W_2) than student translators; in 65.5% and 52.2% of the cases respectively. Student

[3]There are also transitions in the diagonal e.g. $W_1 \rightarrow W_2$ which result from the fact that we have collapsed activities of type 4,5 and 6 into one state. We will ignore them here, since we are not concerned with these transitions in this paper.

translators show more occurrences of TT reading than professionals, which suggests that students aim more often at confirming meaning hypotheses, rather than allocating the cognitive resources directly to writing once a meaning hypothesis has been established. Hvelplund also finds a higher variability in the unit duration of professional translators as compared to student translators. Hvelplund sees this as an indicator for greater ability to adapt to the situation by the professional group.

While Hvelplund investigates the impact of the level of translation expertise on the activity transition probabilities of $S_1 \rightarrow W_2$, we will show below in section 5 that the inner structure of the preceding units (i.e. S_1) themselves seem to determine to some extent the transition to the next state.

4.3 Post-editing styles

Based on a taxonomy that overlaps to some extent with our six activity units, Mesa-Lao (2013) suggests six post-editing steps and develops a minimal model of post-editing with spells out four translation styles. His first two post-editing styles are:

- style$_1$: The post-editor first reads the TT segment, detects an MT error, reads the ST segment, and fixes the MT error.
- style$_2$: The post-editor first reads the ST segment, then the TT segment, detects an MT error, and fixes it.

Translation style$_3$ in Mesa-Lao's taxonomy is a variations of style$_2$ (omit ST reading) and in style$_4$ the post-editor reviews first a previous segment before fixing the MT error. Post-editing style$_1$ seems to be the most preferred among his participants. However, in order to simulate Mesa-Lao's translation styles based on the available data that we have (keystrokes and fixations) and the the four translation states, we cannot know when a translator actually detects an MT error. Skipping the step "detect an MT error" leaves us thus with two post-editing patterns that we can map on sequences of the translation states: style$_1$: $T \rightarrow S \rightarrow W$ and style$_2$: $S \rightarrow T \rightarrow W$. In the following section we reduce these two patterns even further and examine the minimum translation cycles $T \rightarrow W$ and $S \rightarrow W$, which represent the question: what happens before typing?

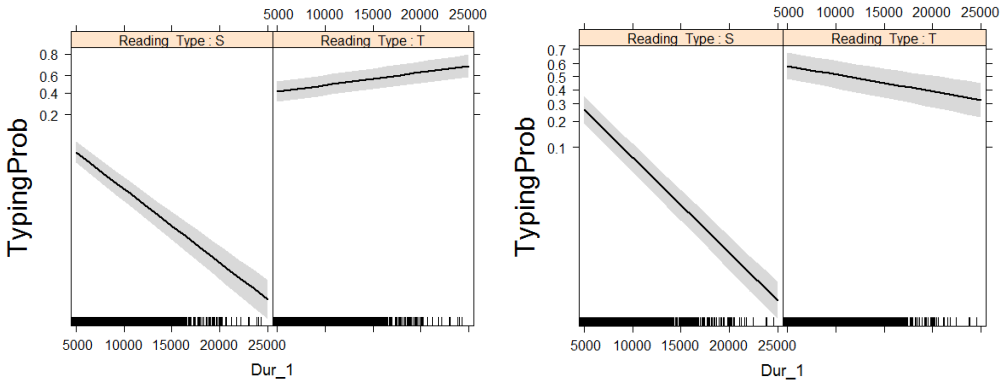
5 Determinants of writing probability

In this section we analyze where and for how long the gaze was observed prior to writing. We will also test to what extent the HCross value (i.e possibility for syntactic choice) of the typed text has an effect on S_1 and T_1 reading duration, prior to typing W_2. The analysis tells us something about the processes which take place between the input (S_1 and T_1 reading), the output (W_2 writing activity) in the cognitive system.

For all the analyses in the present study, R (R Development Core Team, 2014) and the lme4 (Bates et al., 2014) and languageR (Baayen, 2013) packages were used to perform generalized linear mixed-effects models. To test for significance, the R package lmerTest (Kuznetsova et al., 2014) was used. Two separate models (one for post-editing and one for from-scratch translation) with reading duration and HCross as predictors and their interaction with reading type were tested. Both models had participant and target language as random factors.

5.1 The effect of reading duration on writing probability

Increased S_1 reading duration during post-editing (Figure 3a, left) and from-scratch translation (Figure 3b, left) decreases the probability of successive writing (TypingProb). Increased S_1 reading duration thus increases the chances that post-editors and translators engage in successive T_2 reading, instead of writing (W_2). The reason might be due to ST comprehension



(a) **Post-editing** (b) **Translation**

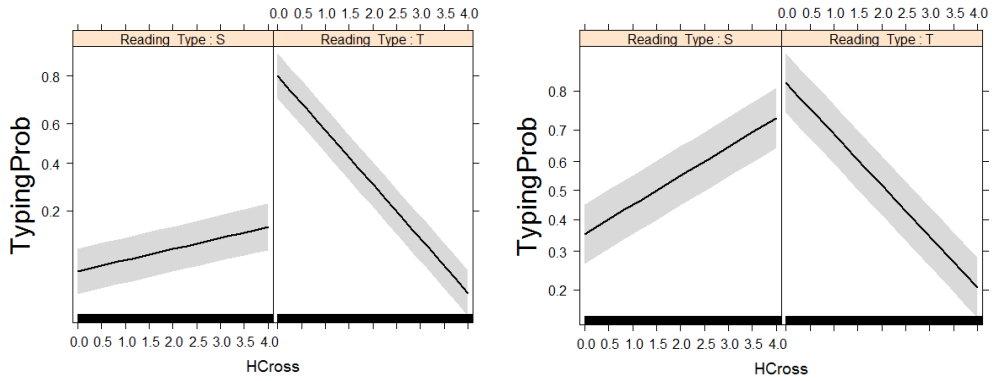
Figure 3: The effect of S_1 and T_1 reading duration (**Dur_1**) on the probability (**TypingProb**) that participants engage in successive writing activity W_2. The gray shadow represents the standard error.

or translation difficulties, which require longer S_1 reading reading times, for both post-editing and from-scratch translation. The more information is processed during ST reading (long S_1 reading), the stronger is the need to first cross check the emerging translation hypothesis with the existing TT, before typing in the translation solution - possibly due to working memory limitations. We thus see more likely a transition $S_1 \rightarrow T_2$ for longer S_1 reading times. That is, a translation hypothesis gathered during S_1 reading needs to be integrated with the existing TT before writing a solution. If the ST information intake is long (long S_1 reading), memory on the status of the TT might first need to be refreshed through (re-newed) TT reading in order for the new solution to be properly integrated. Accordingly, the 16% and 42% of $S_1 \rightarrow W_2$ transitions in post-editing and from-scratch translation respectively (see Table 2) take mainly place if S_1 reading durations are short ($<$ 5000ms, see section 5.3).

Longer T_1 reading activities increase the probability of a successive writing (W_2) for post-editing (Figure 3a, right) but decrease the probability of successive writing for from-scratch translation (Figure 3b, right). This difference in TT reading patterns might be due to the fundamental difference between post-editing and from-scratch translation. In post-editing a TT already exists and some modifications can be made without consultation of the ST. W_2 activities after longer T_1 reading times during post-editing might relate to the correction of (relatively minor) fluency errors which can be corrected without consultation of the TT.

In from-scratch translation, information from the ST needs to be retrieved and integrated with the existing translation in order to continue producing the emerging TT. The longer from-scratch translators read the TT, the more likely they will need to retrieve new information from the ST in order to continue translation production

There were highly significant main effects for reading type, for post-editing (β=4.02, SE=0.06, t=62.54, $p < .001$) and for from-scratch translation (β=2.11, SE=0.04, t=54.65, $p < .001$), such that writing (W_2) was more likely after TT reading (T_1). There were also highly significant main effects for reading duration for post-editing (β=-0.91, SE=0.045, t=-20.38, $p < .001$) and for from-scratch translation (β=-0.68, SE=0.02, t=-33.25, $p < .001$), such that longer reading activities (**Dur_1**) made writing less likely. The interaction between reading type (S_1/T_1) and reading duration (**Dur_1**) was highly significant for post-editing (β=1.07, SE=0.05, t=22.65, $p < .001$) and for from-scratch translation (β=0.56, SE=0.03, t=21.41, $p < .001$).



(a) **Post-editing** (b) **Translation**

Figure 4: The effect of the HCross on the probability that participants type immediately after the reading activity (TypingProb), depending on whether the source (S) or the TT (T) is read prior to the writing event.

5.2 The effect of HCross on writing probability

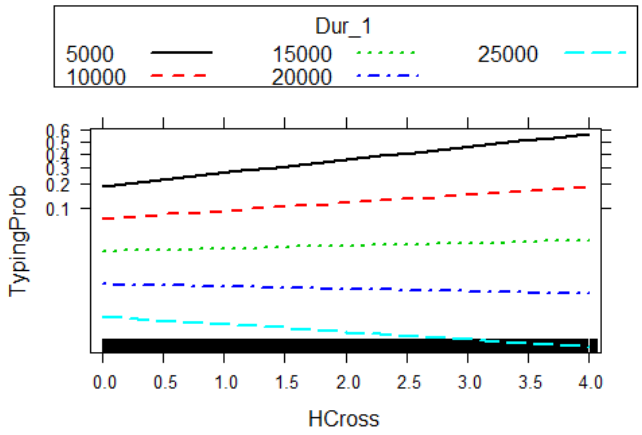
As discussed in section 2, HCross represents the possibility for the translation of a word or phrase to occur in different syntactic positions in the target text segment. HCross is highly correlated with cross-lingual semantic similarity - HTra and HCross correlate to a high degree (r=.79, $p < .001$). The more likely it is that different word orders are realized (high syntactic complexity), the more likely it is that different lexical items are used (high semantic complexity).

For both post-editing (Figure 4a) and from-scratch translation (Figure 4b), HCross had a positive effect on the probability that writing follows ST reading. Thus, higher HCross values increase the probability of a $S_1 \rightarrow W_2$ transition. This effect was more pronounced for from-scratch translation then on post-editing. However, for both post-editing and from-scratch translation, HCross had a negative effect on the probability that writing follows TT reading. Again, this effect was more pronounced for from-scratch translation.

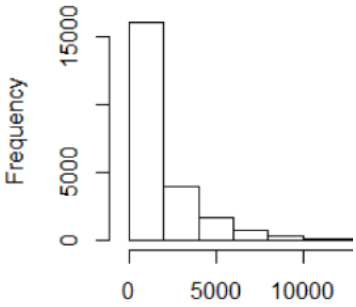
An explanation of this observation might be that items with higher HCross values can be seen as particularly challenging to translate and that solutions for difficult translations emerge during ST reading. The more complex the translation is, i.e. semantically and syntactically less similar, (the less literal), the more likely both post-editors and translators are to refer back to the ST and the less likely they are to type a translation solution immediately after reading the TT.

That is, the 41% and 54% of $T_1 \rightarrow W_2$ transitions in post-editing and from-scratch translation respectively (see Table 2) take preferably place if HCross values are low (the translation is easy). The solutions of more complex translation problems are preferably typed in after S_1 reading.

There were highly significant main effects for HCross, for post-editing (β=0.23, SE=0.02, t=9.73, $p < .001$) and for translation (β=0.41, SE=0.017, t=24.15, $p < .001$), such that writing (W_2) was more likely for higher HCross values. The interaction between reading type (S_1 / T_1) and HCross was highly significant for post-editing (β=-1.34, SE=0.03, t=-44.28, $p < .001$) and for translation (β=-1.12, SE=0.02, t=-49.84, $p < .001$).



(a) Interaction between HCross * Dur_1



(b) Distribution of S_1 duration (Dur_1)

Figure 5: Interaction between the duration (Dur_1) of an S_1 event and the complexity (HCross) of successive translation production W_2 on the probability of a $S_1 \rightarrow W_2$ transition. The typing probability increases with short S_1 reading times and high HCross values. Typing probability decreases with long S_1 reading times and high translation complexity (HCross).

5.3 Interaction of S_1 reading duration and HCross value on W_2 probability

As discussed in the previous sections, the $S_1 \rightarrow W_2$ typing probability during translation depends (among other factors) on the:

- expertise of the translator (section 4.2)
- S_1 reading duration (section 5.1)
- HCross value of the W_2 event (section 5.2)

Figure 5a shows the interaction effect between S_1 reading duration (Dur_1) and the complexity of the translation (HCross) that follows the reading event. In line with the findings discussed in Figures 3a and 3b (left) it shows that short ST reading activities ($< 5000ms$) are followed with high probability by typing events. As shown in Figure 4a and 4b (left) the typing probability is even more likely if the produced translation solution is more complex. This suggests that complex translations are preferably produced immediately after a short ST consultation, presumably to relieve working memory by flushing out probably intermediate and incomplete translation solutions that are later to be revised and thus to avoid building up and keeping more complex structures in mind. In contrast, less complex translation problems may still be integrated with more information gathered during successive TT reading before a typing event occurs.

This trend is reversed for longer ST reading duration, where the typing probability decreases if the translation problem becomes more complex. It suggests that long S_1 reading duration in combination with complex translation problems requires additional T_2 reading, and presumably additional ST-TT integration cycles.

In combination, these observations suggest that difficult translation problems are cross-checked and resolved after reading the ST, while simple translation problems may be rectified after TT reading.

6 General Discussion

According to Dillinger (2014, xi), a key ability for post-editors (and translators) is their ability to compare sentences (and texts) across languages, in terms of both literal meaning and the culturally determined patterns of inference and connotation that different phrasings will entail. Patterns of keystrokes and gaze behavior make it possible to trace the origin of problems translators face to establish equivalence across languages. We have shown that bigrams of translation states, i.e., reading the ST or the TT and writing, constitute minimal and coherent problem identification and solution cycles. The degree of complexity (i.e. syntactic choice) clearly predicts subsequent activities, both during translation and post-editing. Remarkable in this regard is the fact that the effect of word-order choices in the target language (HCross) is similar in both tasks, suggesting that post-editors engage in processes which are not unlike those during from-scratch translation, when the raw MT output is faulty. Both post-editors and translators refer back to the source text when the produced TT is semantically and/or syntactically complex or non-literal. We hope that these minimal and coherent problem identification and solution cycles will constitute the building blocks for a more fully fledged model of both post-editing and from-scratch translation.

References

Baayen, R. H. (2013). languageR: Data sets and Functions with "Analyzing Linguistic Data: A Practical Introduction to Statistics". Technical report.

Bates, D. M., Maechler, M., Bolker, B., and Walker, S. (2014). {lme4}: Linear mixed-effects models using Eigen and S4.

Carl, M. (2012). *Translog-II: a Program for Recording User Activity Data for Empirical Translation Process Research*. Paper presented at The Eighth International Conference on Language Resources and Evaluation.

Carl, M., Schaeffer, M. J., and Bangalore, S. (2016). The CRITT Translation Process Research Database. In Carl, M., Bangalore, S., and Schaeffer, M., editors, *New Directions in Empirical Translation Process Research: Exploring the CRITT TPR-DB*, New Frontiers in Translation Studies, pages 13–54. Springer International Publishing, Cham, Heidelberg, New York, Dordrecht, London:.

Dillinger, M. (2014). Introduction. In O'Brien, S., Winther Balling, L., Carl, M., Simard, M., and Specia, L., editors, *Post-editing of Machine Translation: Processes and Applications*, pages IX–XV. Cambridge Scholars Publishing, Newcastle upon Tyne.

Germann, U. (2008). Yawat: Yet Another Word Alignment Tool. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session*, HLT-Demonstrations '08, pages 20–23, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hvelplund, K. T. (2016). *Cognitive efficiency in translation*, pages 149–170. John Benjamins.

Kuznetsova, A., Christensen, R. H. B., and Brockhoff, P. B. (2014). lmertest: Tests for Random and Fixed Effects for Linear Mixed Effect Models (lmer Objects of lme4 Package). R package version 2.0-6.

Läubli, S. and Germann, U. (2016). Statistical Modelling and Automatic Tagging of Human Translation Processes. In Carl, M., Bangalore, S., and Schaeffer, M. J., editors, *New Directions in Empirical Translation Process Research: Exploring the CRITT TPR-DB2*, pages 155–181.

Mesa-Lao, B. (2013). *Eye-tracking Post-editing Behaviour in an Interactive Translation Prediction Environment*, volume 6, page 541. Lund University.

R Development Core Team (2014). R: A language and environment for statistical computing.

Schaeffer, M. J., Carl, M., and Lacruz, I. (2016). Measuring Cognitive Translation Effort with Activity Units. *Baltic Journal of Modern Computing*, 4(2):331–345.

Fine-Tuning for Neural Machine Translation with Limited Degradation across In- and Out-of-Domain Data

Praveen Dakwale p.dakwale@uva.nl
Informatics Institute, University of Amsterdam

Christof Monz c.monz@uva.nl
Informatics Institute, University of Amsterdam

Abstract

Neural machine translation is a recently proposed approach which has shown competitive results to traditional MT approaches. Similar to other neural network based methods, NMT also suffers from low performance for the domains with less available training data. Domain adaptation deals with improving performance of a model trained on large general domain data over test instances from a new domain. Fine-tuning is a fast and simple domain adaptation method which has demonstrated substantial improvements for various neural network based tasks including NMT. However, it suffers from drastic performance degradation on the general or source domain test sentences, which is undesirable in real-time applications. To address this problem of drastic degradation, in this paper, we propose two simple modifications to the fine-tuning approach, namely multi-objective learning and multi-output learning which are based on the "Knowledge distillation" framework. Experiments on English-German translations demonstrate that our approaches achieve results comparable to simple fine-tuning on the target domain task with comparatively little loss on the general domain task.

1 Introduction

Standard neural MT (Bahdanau et al., 2015) is an end-to-end neural network which allows for easy training of the NMT system without the need to separately train large phrase table and n-gram language models. However, neural methods including NMT, are known to be data-hungry and do not generalize well for rare events. As a result, NMT systems trained only on a specific domain with less available parallel data quickly overfit and do not perform better or even comparable to standard statistical MT approaches (Zoph et al., 2016).

Research in domain adaptation deals with the problem of improving the performance of a model trained on a general domain data over test instances from a new domain. A simple solution to achieve better performance on both domains will be to train the network on the combined in-domain and general domain parallel data. However, as already discussed in (Freitag and Al-Onaizan, 2016) there are two problems with this approach. First, re-training the model from scratch on the combined data is time consuming and second due to the relatively small size of in-domain data, the learned model will be biased towards the general domain and may not perform comparatively on the in-domain test instances. Moreover, this solution is not efficient in real life applications because in situations where a general domain model is already deployed in an application, the original training data may not be available at the production time.

A fast and efficient method for domain adaptation for neural methods is "Fine-tuning" which has also been recently applied for Neural Machine translation (Freitag and Al-Onaizan, 2016; Chu et al., 2017). In fine-tuning, a neural network which is already trained on large general domain data is further trained on the small data available for the new target domain (also called in-domain data). Fine-tuning provides significant improvements as compared to both only in-domain training or only out-of-domain (general domain data) training. This is because pre-initialization with the parameters trained on large data prevents overfitting on the small in-domain data while at the same time, training on the new data performs a complete transform of the parameters space corresponding to the events observed in the new data. However, as a result of this transformation to the new parameter space, performance of the resulting model decreases drastically for the test instances from the general domain as there is no guidance for the general domain events while fine-tuning (Li and Hoiem, 2016). In a real-time application such a degradation of translation performance on either of the tasks is undesirable because even after adaptation one would like to use the model for translating sentences from both domains.

To address this problem of drastic degradation by fine tuning, in this paper, we propose two simple modifications to the fine-tuning approach. Both approaches are based on the "Knowledge distillation" framework of (Hinton et al., 2014) where a smaller "student" network learns to mimic a large "teacher" network by minimizing the loss between the output distributions of the two networks. We are motivated by the idea that new tasks can be learned without degrading performance on old tasks, by preserving the parameters shared between the two tasks and fine-tuning the task specific parameters with respect to the supervision from the corresponding labels or distributions (Li and Hoiem, 2016).

Our first modification is a simple multi-objective learning which involves *fine-tuning* a general domain model on a small in-domain data while at the same time minimizing the loss between the output distributions of the "student" network (the model learned by fine-tuning) and the baseline teacher network (model trained on large general domain data). In our second modification, we propose adding multiple output layers to the "student" network corresponding to the different tasks (domains) and learning task-specific parameters for both domains using only the in-domain data while simply fine-tuning the parameters shared between the two tasks. Our experiments demonstrate that both the proposed approaches multi-objective *fine-tuning* and multi-output *fine-tuning*, achieve translation performance comparable to vanilla fine-tuning on the target domain task and at the same time suffer little degradation on the original general domain task.

In Section 2 we discuss the related work on domain adaptation for traditional statistical MT as well as recent approaches in neural MT. We briefly introduce neural MT architecture of (Bahdanau et al., 2015) and Knowledge distillation framework of (Hinton et al., 2014) in Section 3. We introduce and explain our approaches in section 4 and finally discuss experiments and results in section 5 and 6 respectively.

2 Related work

Domain adaptation for Phrase-based MT has been studied excessively in last few years. The main approaches in domain adaptation involves either data selection (Moore and Lewis, 2010; Axelrod et al., 2011), instance weighting (Matsoukas et al., 2009; Foster et al., 2010) or multiple model interpolation (Nakov, 2008; Bisazza et al., 2011; Sennrich, 2011). The model interpolation is the most effective, however a complex approach, as it requires training multiple models including phrase-tables, re-ordering and language models corresponding to each of the domains and then learning the corresponding interpolation weights. In case of SMT, Wei wang et al. (2012) has goals similar to that of our approach i.e. to adapt an SMT system for newer domain while at the same time preserve the original generic performance. This approach pre-classifies

the incoming sentence and accordingly select the corresponding model weights.

For domain adaptation in neural machine translation, most of the recent research has focused mainly on "fine-tuning" over the additional in-domain data. This is due to the fact that the end-to-end architecture of NMT enables fast and efficient training without complexity of re-training individual component for each domain. The first attempt for applying fine-tuning to NMT was Luong and Manning (2015), where they simply adapted an NMT system trained on large general domain data by further training on a small in-domain data. Recently, (Freitag and Al-Onaizan, 2016) extended the fine-tuning approach by using an ensemble of the baseline general domain model and the fine-tuned model. They proposed that the ensemble provides better translation on new domain as well as retain much of the performance on the general domain. Their experiments using TED talks (Cettolo et al., 2012) as in-domain data demonstrated improvements on the in-domain and less performance drop on the general domain. However, as we will discuss in section 6, our experiments demonstrate that even with an ensemble, the performance of the fine-tuned model still degrades significantly on the general domain task, especially on a target domain such as medical documents for which vocabulary, style and lexical translation probabilities are highly different from source domain of news articles. On the other hand, our approach of using the "baseline supervision" while fine-tuning, not only retains general domain performance better than ensemble for two different target domains but also provides a faster and efficient decoding procedure by avoiding the requirement to compute the two output distributions separately as required in the ensemble approach.

Another recent approach for NMT domain adaptation is the "mixed fine-tuning" by Chu et al. (2017). They proposed to fine-tune the baseline model on a parallel corpus which is mix of the in-domain and general domain corpora instead of only in-domain data. They also perform multi-domain NMT by augmenting all the corpora with domain tags. However, as they pointed out "mixed fine-tuning" takes longer training time than vanilla fine-tuning depending on the size of the "mixed" data. Also, this requires robust data selection heuristics to extract only relevant sentences from the general domain and avoid selection of noise. On the other hand, in our approach we completely remove any dependence on the general domain data while fine-tuning. Similary, Sennrich et al. (2016) adapt to the new domain by back translating the target in-domain sentences, adding the new data to the training corpus and fine-tuning on the extended bitext.

A related line of research to our approach is in computer vision where the supervision from the baseline model based on the "Knowledge distillation" framework has been extensively used in various domain adaptation paradigms such as "Deep domain transfer" (Tzeng et al., 2015), "Knowledge adaptation" (Ruder et al., 2017) and "Learning without forgetting" (Li and Hoiem, 2016). The "Learning without forgetting" approach of Li and Hoiem (2016) is very similar to our approach since they propose to reduce the general domain performance degradation by adding multiple nodes in the output layer of a convolutional neural network and fine-tune the parameters on the in-domain data using the supervision from the baseline model. However, we differ from (Li and Hoiem, 2016) in that we not only apply this method to neural machine translation, but we also experiment by training with multiple objectives as well as multiple output layers instead of simply adding new nodes corresponding to the new classes.

3 Background

3.1 Neural Machine Translation

We employ an NMT system based on Luong et al. (2015) which is a simple encoder-decoder network. The encoder is a multi-layer recurrent network (we use LSTMs) which converts an

input sentence x = [$(x_1, x_2, ..., x_n)$] into a sequence of hidden states [$(h_1, h_2, ..., h_n)$].

$$h_i = f_{enc}(x_i, h_{i-1}) \tag{1}$$

Here, f_{enc} is an LSTM unit. The decoder is another multi-layer recurrent network which predicts a target sequence $y = (y_1, y_2, ...y_m)$. Each word in the sequence is predicted based on the last target word y_{i-1}, the current hidden state of the decoder s_j and the context vector c_j. The context vector c_j in turn is calculated using an attention mechanism Luong et al. (2015) as weighted sum of annotations of the encoder states h_i's.

$$c_j = \sum_{i=1}^{n} \alpha_{ji} h_i \tag{2}$$

where α_{ji} are attention weights corresponding to each encoder hidden state output h_i calculated as follows :

$$\alpha_{ji} = \frac{exp(a(s_{j-1}, h_i))}{\sum_{k=1}^{n} exp(a(s_{j-1}, h_k))} \tag{3}$$

s_j is the decoder hidden state generated by LSTM units similar to the encoder.

$$s_j = f_{dec}(s_{j-1}, y_{j-1}, c_j) \tag{4}$$

Given, the target hidden state and the context vector, a simple concatenation combines the information from both vectors into an attentional hidden state $\tilde{s}_t$.

$$\tilde{s}_t = tanh(W_c[c_t; h_t]) \tag{5}$$

This attentional vector $\tilde{s}_t$ is then projected to the output vocabulary size using a linear transformation and then passed through a softmax layer to produce the output probability of each word in the target vocabulary.

$$p(y_j | y_1, ...y_{j-1}, \mathbf{x}) = softmax(W_s \ \tilde{s}_t) \tag{6}$$

The probability of the sentence is modelled as product of the probability of each target word.

$$p(\mathbf{y}) = \prod_{j}^{m} p(y_j | y_1, ...y_{j-1}, \mathbf{x}) \tag{7}$$

The end-to-end network is trained by maximizing log-likelihood over the training data. The log-likelihood loss is defined as

$$L_{NLL}(\theta) = -\sum_{j=1}^{n} \sum_{k=1}^{|V|} (y_j = k) * log(p(y_j = k | x : \theta)) \tag{8}$$

Where y_j is the output distribution generated by the network at each timestep and *'k'* is the true class label, i.e., the reference target word at each timestep which is selected from a fixed vocabulary *'V'*. The outer summation represent that the total loss is computed as the sum over complete target sequence.

3.2 Knowledge Distillation

Knowledge Distillation is a framework proposed by Hinton et al. (2014) for training compressed "student" networks by using supervision from a large teacher network. Assuming, we have a teacher network with large dimension sizes, trained on a large amount of data, a smaller student network with much smaller dimension sizes can be trained to perform comparable or even better than the teacher by learning to mimic the output distributions of the teacher network on the same data. This is usually done by minimizing the cross-entropy or KL-divergence loss between the two distributions. Formally, if we have a teacher network trained on the same data and with a learned distribution $q(y|x;\theta_T)$, the student network (model parameters represented by θ) can be trained by minimizing the following loss:

$$L_{KD}(\theta,\theta_T) = -\sum_{k=1}^{|V|} KLdiv\Big(q(y|x;\theta_T)\ p(y|x;\theta)\Big) \tag{9}$$

where θ_T is the parameter distribution of the teacher network. Commonly, this loss is interpolated with the log-likelihood loss which is calculated with regard to the target labels for the in-domain data

$$L(\theta,\theta_T) = (1-\lambda)L_{NLL}(\theta) + \lambda L_{KD}(\theta,\theta_T) \tag{10}$$

In order to allow the student network to encode the similarities among the output classes, (Hinton et al., 2014) suggests to generate a smoother distribution called "soft-targets" by increasing the temperature of the softmax of both teacher and student network.

Note that "Knowledge distillation" for model compression has been applied for neural machine translation by (Kim and Rush, 2016). However, we use the supervision from teacher network for domain adaptation while fine-tuning.

4 Domain adaptation with baseline supervision

As described in Section 1, in fine-tuning a model pre-trained on general domain data is further trained on the in-domain data which largely shifts the parameter space of the model to the target domain resulting in a performance degradation on the general domain test instances. We propose that fine-tuning on the target domain using an objective similar to one defined in equation (10) using the supervision from both general and the target domain can avoid the performance degradation on the general domain while achieving performance comparable to vanilla fine-tuning on the target domain. We explore this idea by proposing two modifications to the fine-tuning approach.

4.1 Multi-objective fine-tuning (MCL)

In the first modification, we train the network on a joint objective which includes the supervision provided by the hard target labels in the in-domain data as well as the output distribution of the general domain model on the in-domain data. We believe that with such a joint objective, the network can learn a parameter space common to both domains.

Assuming that we have trained an NMT model on large general domain data, we first record the output distribution of this model on all the in-domain data, then fine-tune this baseline model on the in-domain data by minimizing the log likelihood loss between the target references and the output distribution of the network. However, at the same time, for each sentence, we also minimize the KL-divergence (or cross-entropy) loss between recorded teacher distribution and the distribution produced by the student network as shown in Figure 1. Let the general domain data on which the baseline model is trained be represented by x_{out} and the in-domain

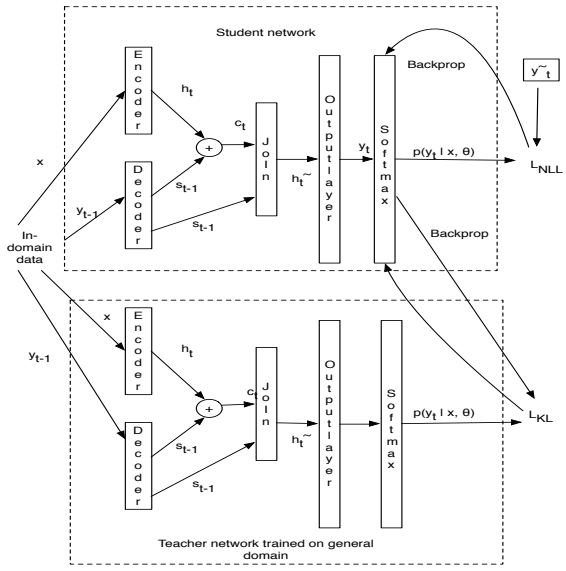


Figure 1: **Multi-objective fine-tuning**. Both the teacher and student network have same architecture. Student network is initialized with parameters trained for teacher network. While fine-tuning parameters of teacher network are frozen.

data be represented by x_{in}, then the final learning objective becomes :

$$\begin{aligned} &L(\theta, \theta_T) = \\ &(1-\lambda) \sum_{k=1}^{|V|} (y_j = k) \times log(p(y_j = k | x_{in} : \theta)) \\ &+ \lambda(- \sum_{k=1}^{|V|} KLdiv\Big(q(y|x;\theta_T)\ p(y|x_{in};\theta)\Big)) \end{aligned} \tag{11}$$

Where θ_T is learned over the general domain data using the standard learning objective in equation 8. Note that as discussed in section 3.2 both distributions here are obtained by increasing the temperature of the softmax as defined in equation 6. This is done by dividing the input by a constant hyper-parameter ϵ

$$p_\epsilon(y|x;\theta) = softmax(\frac{W_s\ \tilde{s}_t}{\epsilon}) \tag{12}$$

Note that Equation 11 clearly indicates that the proposed approach does not require the original data while fine-tuning but only the parameters of the trained baseline model.

4.2 Multiple-output layer fine tuning (MLL)

Though learning on a joint objective as discussed in section 4.1 can reduce the degradation on the general domain task to some extent, a much better solution in order to retain the performance on the old-task could be to preserve task specific parameters corresponding the old-task and at the same time slightly transform parameters shared across the two tasks. Therefore, as a second modification, we propose modifying the baseline model by adding parameter specific to the new task and learning the task-specific parameters with respective learning objectives.

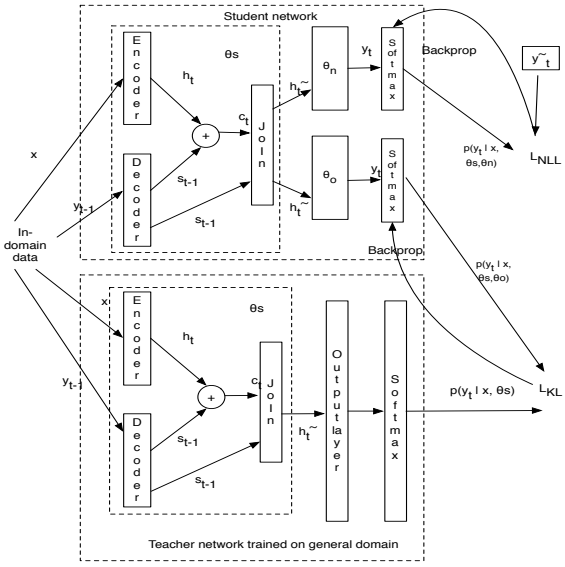


Figure 2: **Multi-output-layer learning**. Student network has two output layers. The additional layer is trained wrt distribution from teacher network

In this work, we consider only the parameters of the final output-layer (W_s as defined in equation 6) of the NMT network as task-specific, while all the parameters corresponding to encoder, decoder, attention mechanism and the concatenation layer in equation 5 are considered to be shared. Let θ_s to be the set of all the shared parameters, θ_o and θ_n to be the task-specific output-layer parameters corresponding to the old (general domain) and new (in-domain) task.

Training the general domain baseline model results in initial-learning of the shared parameters θ_s and the output layer for the out-of-domain task θ_o. For training the in-domain student model, we first modify the network by adding another output layer to the standard NMT network as shown in Figure 2. Similar to multi-objective fine-tuning, we first note the output distribution of the general domain teacher model on the "in-domain" data. Then the shared parameters for the student network corresponding to encoder-decoder and attention mechanism are initialized with θ_s which are learned from the baseline model. Similarly, the parameters of the output layer corresponding the old-domain task are also initialized with parameters θ_o learned on the general domain task. Parameters specific to the in-domain task θ_n are initialized randomly. Then we first train the new parameters θ_n using the ground-truth with the standard log-likelihood objective as defined in equation 8. This is a simple warm-up procedure which enhances the fine-tuning performance (Li and Hoiem, 2016). During this warm-up, we freeze θ_s and θ_n. Finally all the parameters are fine-tuned by minimizing the joint objective. Consider x_n, y_n to be a sentence pair from the in-domain data. Then y_o be the recorded output of the "teacher" model for the new data, i.e.,

$$y_o = q(x_n, \theta_s, \theta_o) \tag{13}$$

For the "student" network, let y'_o and y'_n be the output distributions from the old and new output-layer respectively corresponding to θ'_o and θ'_n

$$y'_o = p(x_n, \theta'_s, \theta'_o) \tag{14}$$

$$y'_n = p(x_n, \theta'_s, \theta'_n) \tag{15}$$

Then similar to equation 11, we define two objectives. First is the standard log-likelihood loss for the shared parameters θ_s and the parameters for new task θ_n with respect to the target references in the in-domain data

$$L_n = -y_n \times log(p(x_n|\theta'_s, \theta'_n)) \tag{16}$$

The second objective is the KL-divergence between the output distribution of the old-layer of the student network with respect to the recorded distribution of the teacher network on the same in-domain data.

$$L_o = -\sum_{k=1}^{|V|} KLdiv\Big(y_o, p(y'_o|x_n; \theta'_s, \theta'_o)\Big) \tag{17}$$

The student network is finally trained on the joint objective defined as:

$$L_{combined} = (1-\lambda)L_n + \lambda L_o \tag{18}$$

While decoding, the output layer corresponding to the domain label of the test sentences is used to compute the output distribution.

5 Experimental settings

5.1 NMT parameters

In all our experiments, we use an NMT system based on Luong et al. (2015) and implemented using the Torch7 deep learning framework. It is a two layer unidirectional LSTM encoder-decoder with a global attention (dot product) mechanism. Both the encoder and decoder have input embedding and hidden layer of size 1000. As it is common practice, we limit the vocabulary sizes to 60k for the source and 40k for the target side. Parameters are optimized using stochastic gradient descent. We set the initial learning rate as 1 with a decay rate of 0.5 for each epoch after 5th epoch. Model weights are initialized uniformly within [-0.02, 0.02]. A dropout value of 0.2 is applied for each layer. The mini-batch size for out-of-domain is fixed to 64, while for each of the in-domain fine-tunings, we use a batch size of 32. We train for a maximum of 20 epochs and decode with standard beam search with beam size of 10. All models are trained on NVIDIA Titan-X (Pascal) devices.

For the in-domain trainings, we use the same vocabulary extracted from the baseline general domain bitext. Note that our multiple-output layer approach allows for use of a different vocabulary (that can be extracted from the new data) for the in-domain fine-tuning. We experimented with different values of interpolation weights λ (as used in equation 11 and 18) and also with temperature ϵ (defined in equation 12) and obtained the optimal values to be 0.9 and 2 respectively for these hyper-parameters for all the experiments.

5.2 Data

We conducted experiments on English-German translation. For all the settings we use WMT-2015 Bojar et al. (2015) training set as the general domain training data. This training set contains approximately 4.2 million parallel sentences from multiple sources Europarl, news commentary and common crawled articles. We constrain the maximum sequence length to be 80 and remove the sentences greater than this length along with the duplicates. Thus we are left with a bitext of size approximately 4 million, out of which we reserve 5000 sentence for perplexity validation and use the rest for training the general domain baseline model. We use newstest15 (official test set for WMT-2015) as general domain test set.

	Epo.	iwslt	newstest'15
Standard NMT baselines			
TED_{only}		17.95	
WMT_{only}		19.46*	18.54*
Combined training (WMT+ TED)		$22.86^{\blacktriangle}$ (+3.4)	$17.57^{\blacktriangledown}$ (-0.97)
Fine-tuning methods			
FT	1	$19.69^{\triangle}$(+0.23)	$13.94^{\blacktriangledown}$(-4.6)
	3	$21.90^{\blacktriangle}$(+2.44)	$12.68^{\blacktriangledown}$(-5.86)
FTEns		$22.90^{\blacktriangle}$(+3.44)	$16.70^{\blacktriangledown}$(-1.84)
Proposed approaches			
MCL	1	$21.77^{\blacktriangle}$(+2.21)	$16.95^{\blacktriangledown}$(-1.59)
	7	$22.19^{\blacktriangle}$(+2.73)	$16.55^{\blacktriangledown}$(-1.99)
MLL	1	$20.26^{\triangle}$(+0.8)	$18.24^{\triangledown}$(-0.3)
	9	$21.70^{\blacktriangle}$(+2.24)	$16.90^{\blacktriangledown}$(-1.64)

Table 1: BLEU scores for different approaches over TED data domain adaptation. iwslt = IWSLT (2011+2012+2013). * represents the baseline setting for these experiments $^{\blacktriangle}/_{\blacktriangledown}$ and $^{\triangle}/_{\triangledown}$ indicates a statistically significant gain/drop at $p < 0.01$ and $p < 0.05$ respectively over the baseline. TED_{only} = only in-domain training, WMT_{only} = only general domain training, FT = Vanilla fine-tuning, FTEns = Ensemble fine tuning, MCL = Multi-objective fine-tuning, MLL = Multi-output-layer fine-tuning

For in-domain training we consider two different domains namely the TED-talks bitext (IWSLT 2013) Cettolo et al. (2012) (approx 170k sentences) and EMEA corpus (Tiedemann, 2009) (approx 200k sentence) which is a parallel text of medical guidelines. From each of these, we reserve 5000 sentence for respective perplexity validation. For the TED talk domain, we use a combination of official test sets for IWSLT 2011, 2012 and 2013 as evaluation set and for EMEA, we reserve a set of 2000 sentences (excluded from training corpus) from the EMEA corpus as evaluation set. Results are reported in terms of case-insensitive BLEU-4 (Papineni et al., 2002). Approximate randomization (Noreen., 1989; Riezler and Maxwell, 2005) is used to detect statistically significant differences.

6 Results

We define multiple baselines to compare the proposed approaches. Firstly, we obtain the BLEU scores for a model trained only on general domain data and note its performance on the general domain (source domain) test set as well as on the in-domain test sets. Similarly, for baseline comparisons we train models only on the small in-domain data for each of the target domains. We compare our approaches to the vanilla fine-tuning and ensemble methods (Freitag and Al-Onaizan, 2016) in terms of the BLEU improvements on the in-domain test sets and degradation on the general domain test-set. Finally, we also train baseline models on the combined general and in-domain bitext and test it on both test sets. Note that this baseline act as ceiling for our approaches as this setting can only be trained when data from both general as well as target domain are available beforehand.

For the domain adaptation over TED data, Table 1 summarizes the BLEU scores over different settings and also the highest and lowest BLEU scores for fine-tuning as well as the proposed approaches. The performance of general-domain-only (WMT_{only}) model (19.46) is higher than that of in-domain-only (TED_{only}) model (17.95) for the iwslt test set. Hence, we consider BLEU scores of the general-domain-only model on the two test sets as our baselines.

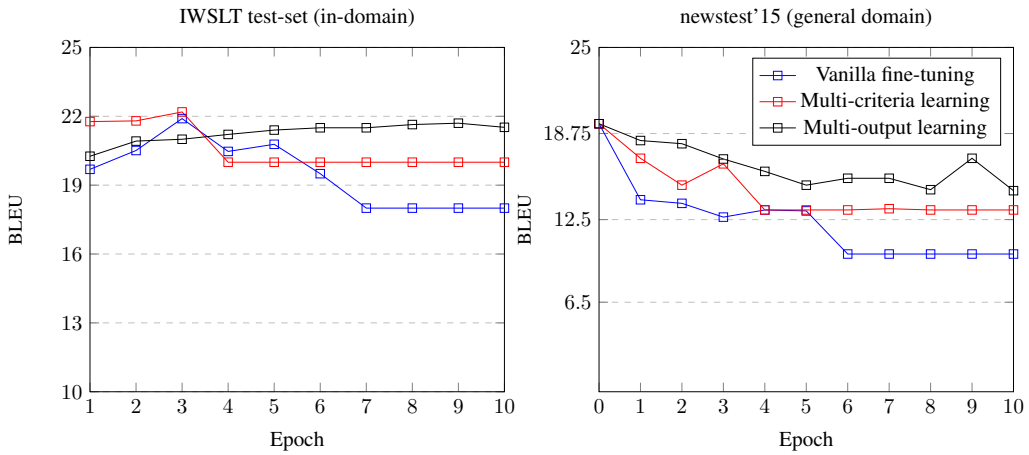


Figure 3: Epoch-wise BLEU (i) improvement over in-domain test set (TED) (ii) degradation over general domain (WMT)

To compare the gradual degradation of the vanilla fine-tuning with the proposed approaches, we report the BLEU scores corresponding to first epoch as well as highest BLEU score achieved for the in-domain test set. The last column shows the corresponding BLEU scores over the general domain test set.

Vanilla fine-tuning improves over the baseline by up to 2.4 BLEU in the third epoch for the target domain task. However, it drops substantially even in the first epoch by 4.6 BLEU on the general domain task and by 5.8 corresponding to third epoch (best model for target domain test-set). On the other hand, for the target domain, multi-objective learning improves over the baseline by up-to 2.7 BLEU (in third epoch) which is slightly better than vanilla fine-tuning while at the same time, performance degradation on the source domain is only -1.99 which is substantially less than that of fine-tuning. Similarly, for multi-output-layer learning, we observe improvements of up to 2.2 BLEU over the baseline on the target domain which is almost equal to fine-tuning and the drop in BLEU over the source domain for the 9th epoch is only 1.64. The ensemble method of (Freitag and Al-Onaizan, 2016) achieves an improvement of 1 BLEU over simple fine-tuning for in-domain test set (iwslt) which is higher than both proposed methods and also suffers similar degradation for general domain (newstest'15). This is due to the fact that though TED talk is considered to be a different domain, the vocabulary and style of the TED data is very similar to that of the general domain data. Also, the model trained on the combined data performs comparable to the best setting i.e, the ensemble method for both test sets. Figure 3 shows the epochwise comparison of BLEU score over source and target domain test sets for TED data domain adaptation. For the source domain fine-tuning performance drops rapidly starting from the first epoch. However, performance degradation of the two proposed approaches on general domain is relatively slow

In conclusion, We observe that for the TED data, our approaches show comparable improvement to fine-tuning while suffering less degradation over source domain. However, the ensemble method of (Freitag and Al-Onaizan, 2016) also demonstrates similar results. Therefore, we repeat our experiments for domain adaptation over the EMEA corpus for which the vocabulary is substantially different from the source domain of news article as compared to the TED data.

Table 3 summarizes our experiments for domain adaptation for EMEA data. First important observation the BLEU score for the EMEA_{only} model is substantially higher (6.5 BLEU

	Epo.	EMEA-test	newstest'15
Standard NMT baselines			
$EMEA_{only}$		20.08 *	
WMT_{only}		13.54	18.54*
Combined training (WMT+ EMEA)		20.84$^{\triangle}$(+0.76)	17.60$^{\blacktriangledown}$(-0.94)
Fine-tuning methods			
FT	1	12.34	7.10$^{\blacktriangledown}$(-11.44)
	6	22.57$^{\blacktriangle}$(+2.49)	4.50$^{\blacktriangledown}$(-14.04)
FTEns		19.43$^{\blacktriangledown}$(-0.65)	13.63$^{\blacktriangledown}$(-4.91)
Proposed approaches			
MCL	1	18.14$^{\blacktriangledown}$(-1.94)	15.50$^{\blacktriangledown}$(-3.04)
	11	**22.50**$^{\blacktriangle}$(+2.42)	**13.29**$^{\blacktriangledown}$(-5.25)
MLL	1	20.6(-0.02)	17.17$^{\blacktriangledown}$(-1.37)
	6	**22.33**$^{\blacktriangle}$(+2.25)	**14.67**$^{\blacktriangledown}$(-3.87)

Table 2: BLEU score for different approaches for EMEA data domain adaptation. * represents the baseline setting for these experiments. EMEA-test = test set for medical domain, $EMEA_{only}$ = training only on in-domain medical data, WMT_{only} = training only on general domain data. Other abbreviations are same as Table 1

points) than the WMT_{only}(this is contrast to the experiments on TED data where the WMT_{only} model performed better than TED_{only} model). Hence, for this set of experiments we consider the general domain performance (20.08) as our baseline. Also, for medical domain adaptation, the model trained on the combined data performs only slightly better than the in-domain baseline over the EMEA test set while comparable to the general domain model on the WMT test set. This implies that this joint model is more biased towards the general domain and hence does not perform better than fine-tuning on the in-domain test set.

Vanilla fine-tuning shows an improvement of up to 2.5 BLEU (in the 6th epoch) over the in-domain-only model on EMEA test-set. However, the drop in performance over the general domain test set is drastic i.e. 11.3 in the first epoch and 14 BLEU in the 6th epoch. Also, though ensemble method suffers less degradation in performance over newstest'15 (4.9 BLEU), it performs slightly lower than the baseline for the EMEA test (-0.6) as compared to vanilla fine-tuning. This implies that a simple ensemble is more biased towards general domain. On the other hand, multi-objective learning performs comparable to fine-tuning on the EMEA test set (+2.4 improvement over baseline) while the drop in general domain performance (-5.25) is not as drastic as vanilla fine-tuning. Similarly, for multi-output-layer approach, while the improvement on EMEA test-set is comparable to both approaches, it shows least drop in performance for newstest'15. Figure 4 shows the comparison of BLEU scores over EMEA test sets and newstest'15 for medical data domain adaptation. Similar to TED data experiments, while fine-tuning performance drops rapidly on newstest'15, for the proposed methods, it is relatively slow.

In conclusion, for the medical domain, proposed methods of multi-objective and multi-output-layer learning show improvements comparable to fine-tuning on the target domain with relatively little loss on the source domain as compared to fine-tuning. On the other hand, ensemble method is biased towards the general domain and fails to add any improvement for the target domain.

Finally, we compare the average decoding time per sentence for ensemble decoding and multi-objective learning in Table 3. The decoding time for ensemble method is twice that of

	Average time(ms)
Fine tuning	0.131
Ensemble decoding	0.277
Multi-objective learning	0.147

Table 3: Average decoding time (in millisecond) on GPU devices per sentence for ensemble decoding and multi-objective learning. Average sentence length for the used test set is 20.9 tokens

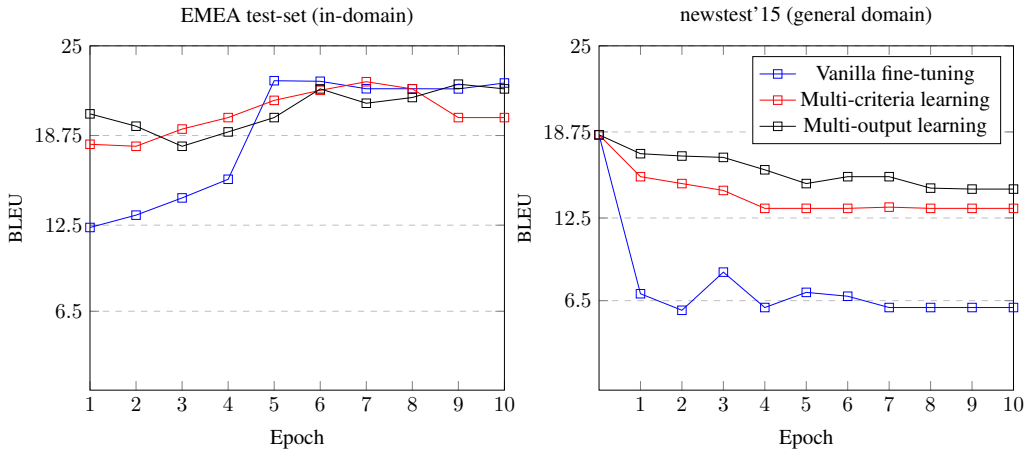


Figure 4: Epoch-wise BLEU (i) improvement over in-domain test set (EMEA) (ii) degradation over general domain test-set (WMT)

fine-tuning due to computation on two different models while for multi-objective learning it is same as that of fine-tuning.

7 Conclusion and future work

In this paper, we proposed two modifications to the well-known fine-tuning method for domain adaptation for neural machine translation in order to retain the performance on the source domain. We observe that both proposed approaches achieve performance comparable to the vanilla fine-tuning while still retaining performance on the source (general) domain. Moreover, the decoding speeds of the proposed methods are same as fine-tuning as compared, while the ensemble method requires almost twice decoding time than the fine-tuning.

In this work, we mainly focused on adapting a general domain model to a new domain. As future work, we plan to experiment by extending the two approaches for iteratively adapting a single model for multiple domains.

Acknowledgments

This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project numbers 639.022.213 and 612.001.218.

References

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11,

pages 355–362. Association for Computational Linguistics.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR*.

Bisazza, A., Ruiz, N., and Federico, M. (2011). Fill-up versus interpolation methods for phrase-based smt adaptation. In *IWSLT*.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., and Turchi, M. (2015). Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Cettolo, M., Girardi, C., and Federico, M. (2012). Wit3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th EAMT Conference, 28-30 May 2012, Trento, Italy*.

Chu, C., Dabre, R., and Kurohashi, S. (2017). An empirical comparison of simple domain adaptation methods for neural machine translation. *CoRR*, abs/1701.03214.

Foster, G., Goutte, C., and Kuhn, R. (2010). Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Cambridge, MA. Association for Computational Linguistics.

Freitag, M. and Al-Onaizan, Y. (2016). Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Hinton, G., Vinyals, O., and Dean, J. (2014). Distilling the knowledge in a neural network. In *NIPS 2014 Deep Learning Workshop*.

Kim, Y. and Rush, A. M. (2016). Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas. Association for Computational Linguistics.

Li, Z. and Hoiem, D. (2016). Learning without forgetting. In *ECCV*.

Luong, M.-T. and Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal. Association for Computational Linguistics.

Matsoukas, S., Rosti, A.-V. I., and Zhang, B. (2009). Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 708–717, Singapore. Association for Computational Linguistics.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10. Association for Computational Linguistics.

Nakov, P. (2008). Improving english-spanish statistical machine translation: Experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 147–150, Stroudsburg, PA, USA. Association for Computational Linguistics.

Noreen., E. W. (1989). *Computer Intensive Methods for Testing Hypotheses. An Introduction*. Wiley-Interscience.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Riezler, S. and Maxwell, J. T. (2005). On some pitfalls in automatic evaluation and significance testing for mt. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Ruder, S., Ghaffari, P., and Breslin, J. G. (2017). Knowledge adaptation: Teaching to adapt. *CoRR*, abs/1702.02052.

Sennrich, R. (2011). Combining multi-engine machine translation and online learning through dynamic phrase tables. In *15th Conference of the European Association for Machine Translation*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany. Association for Computational Linguistics.

Tiedemann, J. (2009). News from opus - a collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. (2015). Simultaneous deep transfer across domains and tasks. In *International Conference in Computer Vision (ICCV)*.

Wei wang, Klaus Macherey, W. M., och, F., and Xu, P. (2012). Improved domain adaptation for statistical machine translation. In *Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas*, San Diego, California.

Zoph, B., Yuret, D., May, J., and Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

Exploiting Relative Frequencies for Data Selection

Thierry Etchegoyhen tetchegoyhen@vicomtech.org
Andoni Azpeitia aazpeitia@vicomtech.org
Eva Martinez García emartinez@vicomtech.org
Vicomtech-IK4, Donostia / San Sebastián, Gipuzkoa, Spain

Abstract

We describe a data selection method for domain adaptation in machine translation, based on relative frequency ratios computed between in-domain and out-of-domain corpora. Our method is compared to a state-of-the-art approach based on cross-entropy differences, outperforming it significantly in terms of data sparseness reduction and BLEU scores on the models created from various data slices. This approach is also shown to either perform significantly better or provide competitive results in terms of perplexity when compared to a method designed to minimise cross-entropy. A novel method to mine unknown words in out-of-domain datasets is also presented, resulting in the best models across the board when used to weight sentences whose similarity to the primary domain is determined by relative frequency ratios. The proposed method is simple, requiring neither external resources nor complex setups, which makes it highly portable across domain adaptation scenarios.

1 Introduction

Data-driven approaches to machine translation, such as statistical machine translation (SMT) (Brown et al., 1990) or neural machine translation (NMT) (Bahdanau et al., 2014), need large volumes of quality bilingual data to be trained effectively. In most scenarios, machine translation systems trained only on available in-domain bilingual corpora face data sparseness issues which hinder on their coverage and accuracy. Identifying useful subsets of out-domain data through automated bilingual data selection has thus become an important method for domain adaptation.

While no fully accurate method has been designed yet to identify subsets of out-of-domain data that are useful and sufficient to improve machine translations models, the main characteristics of the data being sought can be safely assumed to cover two main aspects.

First, the selected out-of-domain sentences should cover lexical and syntactic gaps in the domain, in order to improve the in-domain translation models. This aspect can be controled by measuring the amount of unknown words after incorporating the selected data and by evaluating the impact of the out-of-domain data on the quality of the resulting machine translation models via automated metrics.

Secondly, the selected data should not add confusion to the models, an aspect which can be measured in terms of language model perplexity on either side of the bilingual data. A method that would partially cover lexical and syntactic gaps while also adding significant subsets of data unrelated to the domain at hand would be adding statistical noise to the translation models, thus lowering their accuracy.

These two aspects in combination render the selection task particularly difficult, as the optimal target data should be similar to the in-domain data, in order to reduce the noise added to the models, while also provide enough new material to improve the primary models.

In this paper, we explore the potential of a simple approach based on relative frequency ratios between in-domain and out-of-domain distributions. We evaluate the benefits of our approach in two domain adaptation scenarios featuring large volumes of out-of-domain data and compare it to a state-of-the-art data selection method based on bilingual cross-entropy differences. We show that our approach outperforms data selection based on cross-entropy, achieving significantly better results in terms of translation metrics, while also significantly reducing the amount of out-of-vocabulary words and equating or improving perplexity results.

The remainder of this article is organised as follows: Section 2 summarises related work in bilingual data selection, in particular for domain adaptation; Section 3 describes the proposed approach based on relative frequencies; in Section 4, we present the corpora, models and results of our comparative experiments; finally, in Section 5 we draw conclusions from this work.

2 Related Work

Selecting subsets of bilingual corpora has been a popular approach to create domain-adapted and/or more compact translation systems, see (Eetemadi et al., 2015) for a recent detailed survey. For domain adaptation in particular, the main goal has been to better exploit available parallel corpora, by selecting the minimal subsets of bilingual sentence pairs that maximise the accuracy gains of machine translation systems for a specific domain, where training resources are usually scarce.

Several approaches have been explored over the years for bilingual data selection. TF-IDF weighting has been used for instance by (Lü et al., 2007) for similar sentence identification and weighted training, and by (Eck et al., 2005), who combine it with unseen n-gram frequency scoring to create competitive SMT systems based on smaller training sets. Foster et al. (2010) first rank the out-of-domain sentence pairs according to the perplexity of the in-domain target side language model, then retain the number of top-ranked pair that maximizes the BLEU score on a development set. They further refine the selection process by extending the weight learning approach in Matsoukas et al. (2009), through phrase pair weighting, feature-based measures of the usefulness of phrases and incorporating instance-weighting into a linear combination model.

Perplexity-based methods have figured prominently in work focusing on bilingual data selection. (Foster et al., 2010), for instance, use in-domain target side perplexity to rank out-of-domain sentence pairs and select top-ranked pair that maximize the BLEU score on held-out sets, whereas (Mansour et al., 2011) combine language model and translation model cross-entropy scores to the task of data selection. In (Aydın and Ozgür, 2014), the out-of-domain corpora are ranked according to in-domain perplexity and proper subsets of the data are selected using the vocabulary saturation technique of (Lewis and Eetemadi, 2013). One the most popular data selection methods is that of (Axelrod et al., 2011), who extend work by (Moore and Lewis, 2010) by ranking out-of-domain sentences according to bilingual cross-entropy differences as determined by source and target in-domain and out-of-domain language models.[1] Cross-entropy differences select sentences that are both similar to the in-domain data, and unlike the average out-of-domain data. Generalisations of word-based cross-entropy differences have been proposed by (Axelrod et al., 2015a) and (Axelrod et al., 2015b), improving over the standard approach by means of part-of-speech generalisation and class-based language models.

Whereas most approaches attempt to design optimal similarity measures between domains, (Banerjee et al., 2012) use translation quality to guide data selection. In their approach, batches of out-of-domain data are incrementally added to an existing baseline system, evaluated in terms of translation quality on a development set, and a given batch is selected only if its inclusion improves translation quality. Also not focused on sentence similarity is work by (Daumé III and

[1] We will refer to their approach as Modified Moore-Lewis (MML), although it has received a large variety of acronyms in the literature.

Jagarlamudi, 2011), who address the lexical coverage aspect of supplementary data selection by mining unknown words via canonical correlation analysis. (Gascó et al., 2012) use approximations of in-domain probability distributions and n-gram infrequency scores to achieve significant improvements over the baselines and over random selection. In recent work, (Wong et al., 2016) report significant improvements over perplexity-based selection for Chinese-English, by training recurrent neural networks to select supplementary data.

Selection based on bilingual cross-entropy differences can be considered the *de facto* state-of-the-art approach and is standardly used as baseline by competing approaches. In (Kirchhoff and Bilmes, 2014), the use of submodular functions for data selection obtained minor but statistically significant BLEU score gains over MML, whereas (Peris et al., 2017) achieve slight improvements in terms of BLEU scores via neural network-based classification while using less data. (Banerjee et al., 2013) also compare their data selection method, based on quality estimation, to MML and obtain slightly better BLEU scores while using smaller amounts of data as well. Overall, albeit statistically significant in most reported cases, improvements over MML have been small in terms of automated translation metrics and this method can thus still be considered a strong baseline for comparative evaluations.

Although we focus our work on bilingual data selection, it is worth noting that monolingual data selection for language model adaptation has also been a fruitful approach, explored in several studies. (Mediani et al., 2014), for instance, improve over cross-entropy selection by drawing better samples of out-of-domain data and using word association as a mean to add semantic similarity into the selection process. (Mansour et al., 2011) describe a filtering approach based on combined cross-entropy scores for the language and translation models, and report small but statistically significant improvements over standalone methods. Recently, (Duh et al., 2013) have explored the use of neural language models for data selection, and in particular the advantages of continuous vector spaces over n-gram-based approaches on handling unknown words in out-of-domain corpora. We leave monolingual data selection aside in what follows, although we believe our approach to be worth exploring on these grounds as well, given the results in terms of perplexity and data sparseness reduction described in Section 4.

3 Exploiting Relative Frequencies

By their very nature, perplexity-based approaches tend to favour short out-of-domain sentences that exhibit n-gram distributions close to the primary domain. Although this has been shown to be a fruitful approach in some data selection scenarios, it leaves aside the potential contribution of data that is related to the primary domain while also exhibiting different distributions. In the worst case, perplexity-based methods could select out-of-domain sentences that are already present in the in-domain pool, thus defeating the purpose of increasing model coverage using additional data. Although this is not usually the case with contrasted domains, the main expectation is that machine translation models should benefit more from additional data that cover both lexical gaps and unseen syntactic configurations. In order to test this hypothesis, we design a method that scores out-of-domain sentences according to their similarity to the domain of interest, while not biasing selection towards the in-domain n-gram distributions.

3.1 Relative Frequency Ratios

The approach we propose estimates similarity via relative frequency ratios between the in-domain and out-of-domain data. More specifically, we first compute relative frequencies for each word w in corpus c through token counts C as in Equation 1:

$$\phi_c(w) = \frac{C(w)}{\sum_{i=1}^{|c|} C(w_i)} \tag{1}$$

For each out-of-domain pair (s, t), where s is the set of source words and t the set of target words, the relative frequency score is then computed as the sum of the ratios of in-domain and out-of-domain relative frequencies as in Equation 2, taking the arithmetic mean of the scores for the source and target sentences.

$$rfr(s,t) = \frac{\sum_{i=1}^{|s|} \frac{\phi_d(w_i)}{\phi_o(w_i)} + \sum_{i=1}^{|t|} \frac{\phi_d(w_i)}{\phi_o(w_i)}}{2} \quad (2)$$

In the above equation, ϕ_d and ϕ_o denote the relative frequencies computed on the in-domain and out-of-domain corpora, respectively. To reduce the impact of large differences in terms of sentence length, scoring is applied to the sets of tokens composing each sentence. Out-of-domain words that are not represented in the in-domain corpus are ignored, as the frequency ratio would not be computable in this case.

The metric thus favours sentences with the largest amount of words that are more represented in the in-domain than in the out-of-domain. Additionally, it refrains from ignoring the relative distributions of frequent words, such as function words, under the assumption that all words in a given sentence are important to identify similarity as defined in terms of content, register and style. Finally, the metric remains neutral regarding out-of-in-domain words, as they do not impact the score of a given sentence, and does not favour known n-gram distributions as it is based solely on cumulative word frequency ratios.[2]

3.2 Mining Unknown Words

As previously mentioned, the metric described in the previous section ignores words that are not part of the in-domain vocabulary. Out-of-domain sentences that contain out-of-vocabulary (OOV) words will thus be scored according only to the words in the sentence that do pertain to the in-domain data. This might not be optimal in two respects.

First, large pools of out-of-domain corpora are typically noisy, containing, for instance, sentences in languages other than the expected ones or sequences of corrupt characters. The similarity score in this case would only be determined by the known words, typically punctuation, leaving aside the fact that sentences containing mostly OOV words are not likely to improve the translation models.

Secondly, since adding corpora to the models aims at improving model coverage on both lexical and syntactic grounds, sentences that resemble the in-domain data while also providing new vocabulary, and by extension, phrases, should be favoured over those that are only based on known vocabulary.

Taking these two aspects into account, we aim to promote those sentences that exhibit a reasonable amount of out-of-vocabulary words and minimise the score of those with large amounts of OOV words. In order to do so, we complement the core metric with a weighting scheme related to the percentage of OOV words in each out-of-domain sentence.

Let u be the percentage of out-of-vocabulary items in an out-of-domain sentence, given the in-domain vocabulary. The value u is first assigned a weight according to Equation 3:

$$W(u) = sin(\alpha \cdot u^k) \quad (3)$$

Using this sinusoidal function over percentage of unknown words gives us the expected behaviour: sentences above a given threshold of OOV words will be scored negatively, while the

[2] In additional experiments not reported here, data selection based on relative frequency ratios performed better than log-likelihood-based termhood as in (Gelbukh et al., 2010). We hypothesize that this result is due to the latter both ignoring words that have higher relative frequency in the out-of-domain corpus and to the relative demotion of weaker terms.

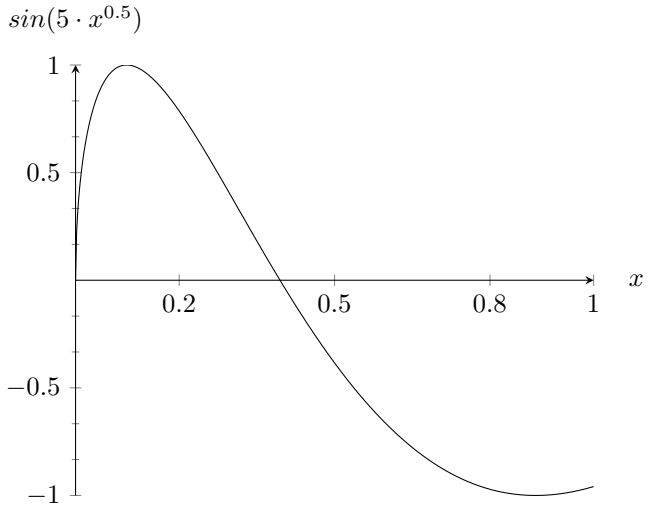


Figure 1: Graph of weighting function for the selected hyper-parameters

higher values from this function will be obtained with a small percentage of unknown words; sentences containing only known words are assigned a value of zero.

The hyper-parameters α and k need to be set empirically, according to how aggressively one wants to mine out-of-vocabulary items. The graph of the function for hyper-parameters α=5 and k=0.5, which were the ones selected for the experiments reported here, is given in Figure 1.

With the selected hyper-parameters, sentences with a percentage of unknown words around 10% will thus be promoted while amounts over 40% will be considered detrimental. The integration of this weighting scheme to the core metric is described in Equation 4.

$$wrfr(s,t) = \frac{exp(W(u_s)) \cdot \sum_{i=1}^{|s|} \frac{\phi_d(w_i)}{\phi_o(w_i)} + exp(W(u_t)) \cdot \sum_{i=1}^{|t|} \frac{\phi_d(w_i)}{\phi_o(w_i)}}{2} \quad (4)$$

Since the core metric is based on relative frequency sums while the weighting scheme ranges over positives and negatives, the values of the W function are mapped to the positive space via exponentiation. Thus, sentences with no unknown words are scored only according to their relative frequency ratios, those with amounts above forty percent will receive a weight between 0 and 1, and the remainder will be favoured with weights above 1. In the next sections, we evaluate the impact of this weighting scheme on the data selection process.

4 Experiments

The experiments described in this section were designed to compare data selection methods in realistic scenarios, where only a fraction of the large out-of-domain data is typically sought. The out-of-domain data, as ranked by each method, were thus sliced from one percent up to twenty percent of the data to perform the evaluations reported here. We compare the two variants of our approach to Modified Moore-Lewis as representative of the state of the art among methods that do not require sophisticated setups and are thus easily portable across domain adaptation scenarios.[3]

[3] MML data selection was performed with the XenC tool (Rousseau, 2013): https://github.com/rousseau-lium/XenC.

LANGS	CORPUS	TRAIN	DEV	TEST
EN-ES	NewsCommentary	207,137	3,003	3,000
EN-FR	EMEA	354,288	500	1000

Table 1: In-domain corpora

LANGS	CommonCrawl	Europarl	UN	POOL
EN-ES	1,814,883	1,842,496	8,079,790	11,661,326
EN-FR	3,065,194	1,826,770	9,142,161	13,864,506

Table 2: Out-of-domain corpora

The data slices used here are similar to those employed by (Axelrod et al., 2011), who experimented with subsets corresponding to 1 time, 2 times and 4 times the size of the in-domain corpus, and by (Axelrod et al., 2012), who opted to select 10% of the out-of-domain data for all of their experiments.[4]

4.1 Corpora

As in-domain corpora, for English-Spanish we used the *NewsCommentary* datasets from the WMT news translation shared task, with *newstest2012* as development set and *newstest2013* as test set; for English-French we used the data from the WMT medical translation task, with EMEA as training set and the *khresmoi-summary* development and test sets.[5]

As out-of-domain corpora, we used three of the available corpora in the aforementioned WMT tasks, namely: *CommonCrawl*, *Europarl* and *UN*. All three corpora were pooled in a single corpus, whose data was then ranked by each method. The statistics for the corpora, after filtering sentences larger than 60 tokens and removing duplicates, are shown in Table 1 and Table 2.

This setup responds to two main goals. First, data selection is applied to a large pool of publicly available out-of-domain data composed of three different sub-domains with varying amounts of noise.[6] This allows for an evaluation of the robustness of each method.

Secondly, the in-domain datasets were selected to be largely different, one covering news commentary and the other medical data, while the out-of-domain data pool remains constant. This provides results in a data selection scenario where the out-of-domain datasets are not pre-selected according to their closeness to the in-domain data. It also allows for a contrastive evaluation of the benefits of the same out-of-domain data for different in-domain corpora.

4.2 Selected Data

The compared methods select different subsets of data at every slice, as shown in Table 3. This is not unexpected given their respective scoring schemes, with short pseudo in-domain sentences being targeted by MML and longer lexically similar sentences by cumulative frequency ratios. The two variants of our approach have a significant amount of common selected sentences on the EN-FR dataset, but more than half of the sentences they select are different up to the 10% slice in EN-ES. This demonstrates the marked impact of the technique we adopted to mine unknown words on the selection of sentences deemed similar by their relative frequency ratios.

The results in Table 3 also show that the two adaptation scenarios differ markedly with

[4]Note that the dichotomic search for optimal slices, computed by the XenC tool using perplexity scores on held-out sets, identified best points at 1% and 14% for the English-French and English-Spanish datasets, respectively.

[5]All datasets are available at http://www.statmt.org/wmt13/ and http://www.statmt.org/wmt14/.

[6]The *CommonCrawl* corpus contains large sections of noisy data, for instance.

LANGS	METHODS	1PCT	2PCT	5PCT	10PCT	20PCT	30PCT	40PCT	50PCT
EN-ES	MML-RFR	11.72	15.88	23.86	32.59	44.12	52.88	60.59	67.91
	MML-WRFR	5.16	7.19	12.17	19.12	30.85	41.39	51.30	60.79
	RFR-WRFR	44.81	43.72	45.42	49.65	57.51	64.40	70.81	77.13
EN-FR	MML-RFR	24.55	24.30	24.72	27.31	34.45	42.08	49.90	57.95
	MML-WRFR	21.09	21.30	22.08	24.88	32.13	39.94	48.04	56.49
	RFR-WRFR	84.20	83.66	82.59	82.59	83.70	85.04	86.39	87.81

Table 3: Percentage of common selected sentence pairs per data slice

LANGS	METHOD	1PCT	2PCT	5PCT	10PCT	20PCT	30PCT	40PCT	50PCT
EN-ES	MML	17.5	19.3	21.6	23.2	24.6	25.4	25.8	26.1
	RFR	40.11	40.03	39.18	37.90	36.03	34.63	33.44	32.30
	WRFR	39.41	40.04	39.85	38.86	37.07	35.52	34.08	32.70
EN-FR	MML	14.2	15.7	17.9	19.7	21.8	23.0	23.7	24.2
	RFR	28.44	29.72	31.86	33.14	33.56	33.10	32.31	31.35
	WRFR	29.77	31.04	33.12	34.19	34.29	33.60	32.63	31.53

Table 4: Average source sentence length per data slice

respect to the distribution of similar sentences in the out-of-domain corpus. Whereas there is a significant portion of similar material in the in-domain and out-of-domain data for EN-ES, due to the presence of the *Europarl* and *UN* corpora, for EN-FR the amount of out-of-domain data related to the medical domain is sparser. This produces the expected larger selection differences between methods in the first case as compared to the second one.

As previously noted, the methods are expected to differ in terms of length, given their respective scoring schemes. The comparative data shown in Table 4 confirm this expectation, with the two methods based on relative frequency ratios selecting sentences that are on average double the length of those selected by MML in the first three slices. Length differences tend to reduce in larger slices, although the perplexity-based approach still tends to select shorter sentences overall.

All three methods select data that seem related to the in-domain at first glance, as illustrated in Table 5 with examples of the type of out-of-domain sentences uniquely selected by each method in their respective 1% slices. In the next sections, we measure more precisely how related the selected data are to the in-domain in terms of capturing out-of-vocabulary words, perplexity and automated translation metrics.

4.3 Unknown Words

One of the main motivations for an approach based on cumulative frequency ratios is its selection of longer sentences similar to the in-domain, which is meant to increase the amount of unknown words that can be captured, indirectly in the case of RFR and directly in the case of WRFR.

To evaluate the differences between the three methods in terms of increasing in-domain coverage, we measured the number of out-of-vocabulary items a posteriori on the test sets for each data slice. Figure 2 shows the results on the source side in EN-ES.

For this language pair, the amount of OOV items under MML is more than double that of WRFR, and nearly double that of RFR, for the lower slices. At the 1% mark in EN-ES, for instance, the slices contain 2669, 1529 and 1146 unknown words when selected by MML, RFR and WRFR, respectively. The amounts of OOV words only start to be similar around the 50

LANGS	METHOD	SENTENCES
EN-ES	MML	where are we heading ? — trillions of dollars more are waiting in the wings . — the implications are dire .
	RFR	the assumption that only an enlightened minority is in a position to respect human rights and freedoms . — greenhouse gas emissions can be cut through the use of nuclear energy , clean coal and low carbon-emitting renewable energies . — coupled with extensive deregulation of financial markets and excess liquidity , these imbalances encouraged investors to engage in leveraged risk-taking in search of profits .
	WRFR	during that period , their debt actually increased from $ 618 billion in 1980 to $ 3.25 trillion in 2006 . — Mr. Snowden (United States of America) said that the Commission for Sustainable Development had galvanized action and helped shape the agendas of a wide range of organizations around the world . — there has been a temptation for the West – Europe and the United States – to stress continuity and so-called stability .
EN-FR	MML	avoid contact with skin , eyes or clothing . — the unused portion should be discarded . — peel open the package with dry hands and place the tablet on your tongue .
	RFR	in terms of public health , the environmental impact of the new medicinal products should be assessed . — antiretroviral treatment can be effective only if it is administered and monitored by health professionals working in a well-functioning national health system . — finally , it recognises the need for studies on vaccines and anti-viral medications that are independent of the pharmaceutical industry , including with regard to the monitoring of vaccination coverage .
	WRFR	during the final process , an operator peers through a microscope at the die surfaces , polishing them carefully with a diamond abrasive tool head that is vibrated by supersonic waves . — concentrations of petroleum contaminants in fish and crab tissue , as well as contamination of shellfish could have potentially significant adverse effects on health . — the first three , namely glycerine , brake fluid and anti-freeze , are considered to present the most extreme incompatibility with calcium hypochlorite .

Table 5: Uniquely selected English sentences in 1% slices

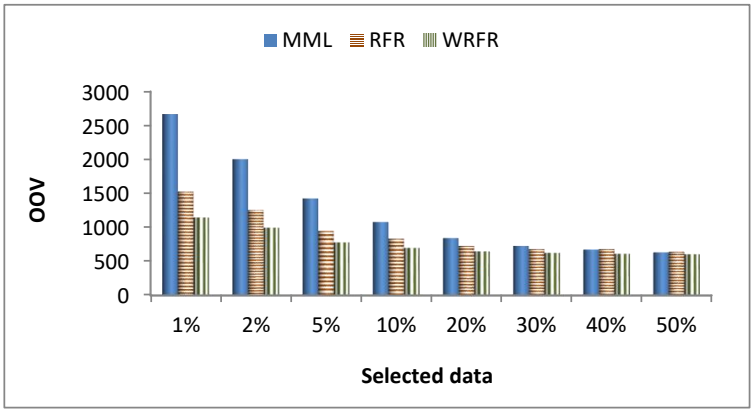


Figure 2: Source out-of-vocabulary words per English-Spanish data slice

percent mark, although WRFR still captures more unknown words in all cases.

The results for the source side in EN-FR are shown in Figure 3.

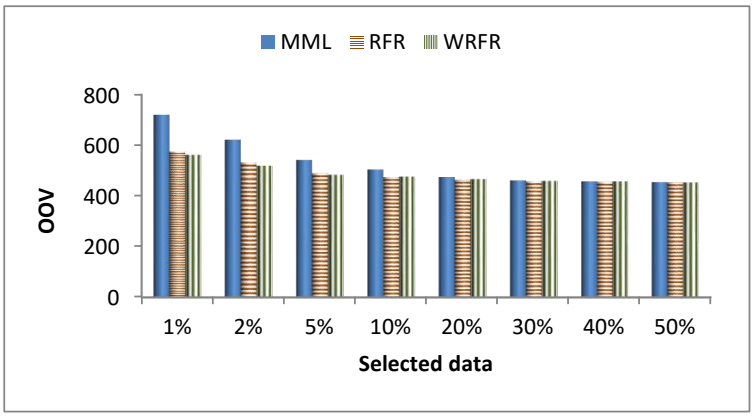


Figure 3: Source out-of-vocabulary words per English-French data slice

For this language pair, the tendencies are similar, with MML being markedly outperformed by both RFR and WRFR, and the latter being the best of all three methods in terms of selecting out-of-domain data that reduce data sparseness.

4.4 Perplexity

As seen in the previous section, the compared approaches differ significantly in terms of selected data, in terms of both average length and amount of OOV items they capture. In addition to these measures, it is important to evaluate the increase or reduction in their respective statistical modelling of sequences. Table 6 indicates the perplexities, including OOV words in the computation of entropy, obtained by each method on the respective test sets after training language models on each data slice.

Since MML is designed to target those sentences that have low in-domain perplexity and high out-of-domain perplexity, one could expect this method to significantly outperform methods based on relative frequency ratios, which make no attempt at minimizing perplexity. As shown above, this is not the case, with both RFR and WRFR significantly outperforming MML

LANG	METHOD	1PCT	2PCT	5PCT	10PCT	20PCT	30PCT	40PCT	50PCT
ES	MML	335.55	295.46	249.95	217.95	**196.59**	**188.95**	**186.60**	**186.60**
	RFR	281.53	252.06	224.52	210.23	201.26	198.49	197.38	197.17
	WRFR	**257.67**	**232.76**	**211.72**	**202.32**	197.93	197.15	196.85	196.79
FR	MML	**151.90**	**147.55**	**151.63**	**161.38**	**175.67**	**187.42**	**196.67**	**203.95**
	RFR	153.63	154.66	163.54	173.54	187.74	197.88	205.13	210.47
	WRFR	157.64	158.75	166.89	177.64	191.15	200.44	207.20	212.14

Table 6: Target language perplexity with OOV per data slice

up to the 10% slice in EN-ES. For EN-FR, the MML approach provides better results on all slices, but only marginally so when compared to the differences obtained in EN-ES.

With both RFR and WRFR outperforming MML in terms of OOV coverage, it could be hypothesised that the competitive results in terms of perplexity are largely due to differences in the amount of captured unknown words. To evaluate this specific point, we computed perplexity using the same language models but ignoring OOV words, with the results shown in Table 7.

LANG	METHOD	1PCT	2PCT	5PCT	10PCT	20PCT	30PCT	40PCT	50PCT
ES	MML	221.38	211.69	193.56	177.30	**165.77**	**162.42**	**161.72**	**161.72**
	RFR	217.07	202.41	186.75	178.65	173.77	172.45	172.28	172.57
	WRFR	**211.53**	**196.18**	**182.75**	**176.67**	173.97	173.61	173.51	173.38
FR	MML	**116.81**	**116.01**	**121.36**	**129.67**	**140.90**	**149.75**	**157.47**	**163.06**
	RFR	123.81	125.72	132.94	140.82	151.35	159.15	164.47	168.46
	WRFR	127.86	128.82	135.70	144.10	154.29	161.20	166.20	169.95

Table 7: Target language perplexity without OOV per data slice

The tendencies observed for perplexity including all words are maintained, with RFR and WRFR obtaining the best scores on the first slices in EN-ES and MML obtaining lower perplexities across the board for EN-FR. The differences between the methods are less marked in this case, due to ignoring out-of-vocabulary items in the computation of perplexity.

Overall, the two variants based on relative frequencies perform well in terms of perplexity, either outperforming the perplexity-minimising MML approach or reaching comparable results.

4.5 Extrinsic Evaluation

Finally, we performed extrinsic evaluations using SMT models trained on the in-domain and out-of-domain corpora, as there exist well-established methods to perform domain adaptation with said models. All translation models are phrase-based (Koehn et al., 2003), trained using the Moses toolkit (Koehn et al., 2007) with default hyper-parameters and phrases of maximum length 5. The phrase tables were pruned according to statistical significance (Johnson et al., 2007) and the parameters of the log-linear models were tuned with MERT (Och, 2003). All language models are of order 5, trained with the KENLM toolkit (Heafield, 2011). The individual in-domain and out-of-domain translation models were then combined by filling up the in-domain phrase table with out-of-domain phrases, with a binary feature denoting the origin of each phrase (Bisazza et al., 2011).

We did not perform additional extrinsic evaluations using neural machine translation models for this work. Although this could provide valuable additional information, domain adaptation with NMT is an ongoing research activity where current approaches have certain limitations. One of the main methods currently employed is that of specialisation, where a network trained

MODEL	100PCT	1PCT	2PCT	5PCT	10PCT	20PCT
NEWSCOM	23.285	_	_	_	_	_
POOL	27.746	_	_	_	_	_
RAND	_	24.065	24.246	25.194	26.273	27.01
MML	_	23.637	24.121	24.999	26.101	26.878
RFR	_	24.547 † ‡	25.102 † ‡	26.0 † ‡	26.563 † ‡	27.166 ‡
WRFR	_	**24.823** † ‡ ∗	**25.458** † ‡ ∗	**26.142** † ‡	**26.914** † ‡ ∗	**27.258** † ‡

Table 8: BLEU scores per data slice for English-Spanish

MODEL	100PCT	1PCT	2PCT	5PCT	10PCT	20PCT
EMEA	27.099	_	_	_	_	_
POOL	37.958	_	_	_	_	_
RAND	_	31.564	31.747	33.234	34.52	37.43
MML	_	33.695 †	34.805 †	35.907 †	36.539 †	37.43
RFR	_	34.791 † ‡	**35.48** † ‡	**35.979** †	**37.438** † ‡ ∗	37.276
WRFR	_	**35.124** † ‡ ∗	**35.325** † ‡	**36.298** †	36.987 † ‡	37.268

Table 9: BLEU scores per data slice for English-French

on generic data is subsequently extended with additional in-domain data (Crego et al., 2016). As it stands, this method requires the new data to be constrained to the vocabulary of the already trained network, which prevents a direct contribution of in-domain vocabulary. This specific issue is typically mitigated via external dictionaries along with a copy mechanism for words that are not part of the generic vocabulary, a working solution which does not however allow for a complete modelling of the in-domain data. Adopting a reversed approach would result in training an in-domain model and add the selected out-of-domain data, as is typically done in SMT domain adaptation. However, the networks would specialise towards the selected out-of-domain data in this case, which would not provide the expected domain adaptation results. A third approach would be to train several models from scratch using a combination of all the in-domain data along with each slice of selected out-of-domain data, a highly computationally expensive approach since each addition of selected data slices would require the training of an entire network.

The time-tested SMT-based approach we chose for our experiments has the advantage of not putting internal restrictions on the available vocabulary and allows for a straightforward comparison between the different data selection approaches. We thus leave additional NMT-based contrastive experiments for future work, noting that evaluating the contribution of selected portions of out-of-domain data, as determined by each one of the compared methods, on NMT models, would undoubtedly provide interesting additional results.

In addition to the models trained on each slice as selected by the three compared methods, for both scenarios we trained a POOL model by combining the in-domain model with a model trained on all out-of-domain data, and used randomly sampled data to train a random baseline (RAND). The comparative results for the two domain adaptation scenarios in terms of BLEU scores (Papineni et al., 2002) are shown in Tables 8 and 9 for English-Spanish and English-French, respectively.[7]

[7]Statistical significance was measured using the paired bootstrap resampling test of (Koehn, 2004) over average BLEU scores. † indicates statistical significance, at $p < 0.05$, as computed between a given model and the random baseline; ‡ between RFR or WRFR and MML; and ∗ between RFR and WRFR.

Overall, the RFR and WRFR approach outperformed MML across the board, with results exhibiting no statistically significant differences between the three models obtained only at the 20% slice mark in EN-FR. Using only 1% of the data, the WRFR approach improves over MML by 1.2 BLEU points for EN-ES and by 1.4 for EN-FR. Given the usually minor improvements obtained by alternatives against MML, these results are indicative of the ability of approaches based on relative frequencies to select useful data that help reach significant improvements of machine translation models.

Note that, for EN-ES, there is no statistically significant difference between MML and the random baseline, although all methods perform better than random selection throughout in EN-FR. This difference might be attributable to the fact that MML tends to select already known material, which is more likely to be selected in this out-of-domain pool when the in-domain contains news-related data. Thus, the selected data bring less new and useful data than is the case for EN-FR, where there is a wider gap between medical in-domain data and the average data in the out-of-domain pool.[8]

Also interesting to note is the fact that no model performs better than the ones created with all out-of-domain data. Note that several reports of models trained on a subset of the data having outperformed the reference models trained on all data indicated such results when using larger slices than the ones reported here (see e.g., Banerjee et al. (2012); Wong et al. (2016)); in other cases, the best results do not outperform the larger models (see, e.g., Peris et al. (2017)). Results on these grounds are also largely dependent on the volumes and nature of out-of-domain data being used; in our case, the pools are on the larger side of the reported experimental scales and contain merged data from different domains, which renders the task more difficult for any data selection method. In any case, the methods evaluated here already reach results that are close to those obtained using all the available data, while using only a fraction of the data, which is one of the main reasons to apply data selection.

5 Conclusion

We described a data selection method for domain adaptation in machine translation, based on relative frequency ratios computed between in-domain and out-of-domain corpora. Our method was compared to a state-of-the-art approach based on cross-entropy differences, outperforming it in terms of data sparseness reduction and BLEU scores on the models created from various data slices. Although not meant to minimise perplexity, our approach was shown to either perform significantly better with fewer data or provide competitive results.

A novel method to mine unknown words in out-of-domain datasets was also presented, which resulted in the best models across the board when used to weight sentences whose similarity to the primary domain was determined by relative frequency ratios. This empirical method can be applied to other scenarios as well, where the goal is to target sentences according to the desired amount of unknown words.

The proposed method is simple, requiring neither external resources nor complex setups, which makes it highly portable across domain adaptation scenarios. In future work, we will pursue improvements and comparative evaluations of the presented methods, in particular with neural machine translation models, where the comparatively larger amounts of useful data retrieved by the method we described might also contribute to increase model accuracy.

[8] Since MML depends on sampling the out-of-domain data in similar proportion to the size of the in-domain, different samples might give different results, especially on large datasets. Several samples could be drawn from the same out-of-domain datasets, a functionality that is provided by the XenC toolkit. However, results along these lines have not been fully explored, to the best of our knowledge, and we opted to use the MML method in its standard variant with single sampling.

References

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics.

Axelrod, A., He, X., Resnik, P., and Ostendorf, M. (2015a). Data selection with fewer words. pages 58–65.

Axelrod, A., Li, Q., and Lewis, W. D. (2012). Applications of data selection via cross-entropy difference for real-world statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 201–208.

Axelrod, A., Vyas, Y., Martindale, M., Carpuat, M., and Hopkins, J. (2015b). Class-based n-gram language difference models for data selection. In *Proceedings of the 12th International Workshop on Spoken Language Translation*.

Aydın, B. and Ozgür, A. (2014). Expanding machine translation training data with an out-of-domain corpus using language modeling based vocabulary saturation. In *Proceedings of the Eleventh Conference of the Association for Machine Translation in the Americas (AMTA)*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*.

Banerjee, P., Naskar, S. K., Roturier, J., Way, A., and van Genabith, J. (2012). Translation quality-based supplementary data selection by incremental update of translation models. In *Proceedings of COLING 2012: Technical Papers*, pages 149–166.

Banerjee, P., Rubino, R., Roturier, J., and van Genabith, J. (2013). Quality estimation-guided data selection for domain adaptation of smt. *MT Summit XIV: proceedings of the fourteenth Machine Translation Summit*, pages 101–108.

Bisazza, A., Ruiz, N., Federico, M., and Kessler, F.-F. B. (2011). Fill-up versus interpolation methods for phrase-based smt adaptation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 136–143.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85.

Crego, J., Kim, J., Klein, G., Rebollo, A., Yang, K., Senellart, J., Akhanov, E., Brunelle, P., Coquard, A., Deng, Y., et al. (2016). Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Daumé III, H. and Jagarlamudi, J. (2011). Domain adaptation for machine translation by mining unseen words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 407–412. Association for Computational Linguistics.

Duh, K., Neubig, G., Sudoh, K., and Tsukada, H. (2013). Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 678–683.

Eck, M., Vogel, S., and Waibel, A. (2005). Low cost portability for statistical machine translation based on n-gram coverage. In *Proceedings of MTSummit X*.

Eetemadi, S., Lewis, W., Toutanova, K., and Radha, H. (2015). Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29(3-4):189–223.

Foster, G., Goutte, C., and Kuhn, R. (2010). Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 451–459, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gascó, G., Rocha, M.-A., Sanchis-Trilles, G., Andrés-Ferrer, J., and Casacuberta, F. (2012). Does more data always yield better translations? In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 152–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gelbukh, A., Sidorov, G., Lavin-Villa, E., and Chanona-Hernandez, L. (2010). Automatic term extraction using log-likelihood based comparison with general reference corpus. In *International Conference on Application of Natural Language to Information Systems*, pages 248–255. Springer.

Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.

Johnson, J. H., Martin, J., Foster, G., and Kuhn., R. (2007). Improving translation quality by discarding most of the phrasetable. In *In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 967–975.

Kirchhoff, K. and Bilmes, J. (2014). Submodularity for data selection in statistical machine translation. In *Proceedings of the 2014 Conference on Empricial Methods in Natural Language Processing*.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 388–395.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL*, pages 177–180. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics-Volume 1*, pages 48–54. Association for Computational Linguistics.

Lewis, W. and Eetemadi, S. (2013). Dramatically reducing training data size through vocabulary saturation. In *Proceedings of the Eigth Workshop on Statistical Machine Translation*, pages 281–291.

Lü, Y., Huang, J., and Liu, Q. (2007). Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 343–350.

Mansour, S., Wuebker, J., and Ney, H. (2011). Combining translation and language model scoring for domain-specific data filtering. In *Proceedings of the International Workshop on Spoken Language TTranslation*, pages 222–229.

Matsoukas, S., Rosti, A.-V. I., and Zhang, B. (2009). Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 708–717, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mediani, M., Winebarger, J., and Waibel, A. (2014). Improving in-domain data selection for small in-domain sets. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT2014)*.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 220–224, Stroudsburg, PA, USA. Association for Computational Linguistics.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Peris, Á., Chinea-Rios, M., and Casacuberta, F. (2017). Neural networks classifier for data selection in statistical machine translation. In *Proceedings of the 20th Annual Conference of the European Association for Machine Translation*.

Rousseau, A. (2013). Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, 100:73–82.

Wong, D. F., Lu, Y., and Chao, L. S. (2016). Bilingual recursive neural network based data selection for statistical machine translation. *Knowledge-Based Systems*, 108:15–24.

Low Resourced Machine Translation via Morpho-syntactic Modeling: The Case of Dialectal Arabic

Alexander Erdmann ae1541@nyu.edu
Computational Approaches to Modeling Language (CAMeL) Lab,
New York University Abu Dhabi
Department of Linguistics, Ohio State University

Nizar Habash nizar.habash@nyu.edu
Computational Approaches to Modeling Language (CAMeL) Lab,
New York University Abu Dhabi

Dima Taji dima.taji@nyu.edu
Computational Approaches to Modeling Language (CAMeL) Lab,
New York University Abu Dhabi

Houda Bouamor hbouamor@cmu.edu
Department of Computer Science, Carnegie Mellon University Qatar

Abstract

We present the second ever evaluated Arabic dialect-to-dialect machine translation effort, and the first to leverage external resources beyond a small parallel corpus. The subject has not previously received serious attention due to lack of naturally occurring parallel data; yet its importance is evidenced by dialectal Arabic's wide usage and breadth of inter-dialect variation, comparable to that of Romance languages. Our results suggest that modeling morphology and syntax significantly improves dialect-to-dialect translation, though optimizing such data-sparse models requires consideration of the linguistic differences between dialects and the nature of available data and resources. On a single-reference blind test set where untranslated input scores 6.5 BLEU and a model trained only on parallel data reaches 14.6, pivot techniques and morpho-syntactic modeling significantly improve performance to 17.5.

1 Introduction

Arabic is widely spoken and highly diglossic, with Modern Standard Arabic (MSA) representing the high register shared across the Arab World in educated circles. In contrast, the many spoken dialectal Arabic varieties (DA) are somewhat if not entirely mutually unintelligible, e.g., Moroccan and Kuwaiti. Chiang et al. (2006) compare the linguistic variation among Arabic dialects to that among Romance languages, indicating the need for machine translation (MT)

between these dialects. However, while much MT research has been devoted to translating between Romance languages (Corbí Bellot et al., 2005; Armentano-Oller et al., 2006; Koehn et al., 2009), we are aware of only one work on Arabic DA-to-DA MT (Meftouh et al., 2015). It deals mainly with Maghrebi dialects and utilizes only a small parallel corpus.[1] This work focuses on the Egyptian and Levantine dialects, leveraging various available resources such as a morphological analyzer and additional monolingual and multilingual data.[2] Compared to other dialects, Egyptian and Levantine's wider range of available data/resources allows us to evaluate more MT approaches using different combinations of these data/resources. Thus, in future work on DA pairs which may not have the same data/resources, we can tailor MT systems based on this paper's findings.

The main challenge in developing DA-to-DA MT systems is the lack of data. While many Romance languages are official languages with written standards, naturally occurring in parallel corpora like the European Parliament (Koehn, 2005), DA has no official status and was rarely written until the advent of social media.[3] The recent release of the first parallel multi-dialectal corpora (Bouamor et al., 2014; Meftouh et al., 2015) has enabled seminal, albeit low-resource MT experiments. We present some shortcomings of these corpora and introduce an in-house, under-development corpus. Then we explore different means of leveraging external resources, e.g., Egyptian-to-English and Levantine-to-English data and an Egyptian tokenizer and morphological analyzer (Habash et al., 2012b; Maamouri et al., 2014; Pasha et al., 2014). We conduct experiments in a range of data-sparse settings and show the effect of morpho-syntactic features on the DA-to-DA MT performance. Our approach can be extended to other DA pairs and other closely related languages and dialects (Tyers et al., 2017).

2 Related Work

An increasing amount of research has been conducted on dialectal Arabic NLP; however, most dialectal MT efforts translate from DA to MSA or English. The only other DA-to-DA work we are aware of focuses on manipulating language model smoothing parameters to optimize data sparse MT performance (Meftouh et al., 2015).

2.1 Dialectal Arabic Machine Translation

While only Meftouh et al. (2015) have evaluated DA-to-DA MT, many others have addressed MT between DA and other languages. Zbib et al. (2012) attempted to translate from Egyptian and Levantine to English and found that pivoting through better resourced MSA was not useful due to register and domain differences. MSA, the higher register, is rarely used to discuss the day-to-day matters frequently treated with DA, causing a domain mismatch. However, several approaches have since presented alternative results (Sawaf, 2010; Salloum and Habash, 2011, 2012; Sajjad et al., 2013; Durrani et al., 2014). These use rule-based or hybrid methods to identify mappings from DA to MSA before translating to a target language (usually English). Additionally, Tachicart and Bouzoubaa (2014) report results on adapting an approach designed for MSA to Moroccan translation to translate in the inverse direction (Moroccan to MSA).

2.2 Dialectal Arabic Data

Several newly developed corpora have facilitated the recent surge in dialectal NLP work.

[1] Maghrebi dialects are those spoken in Morocco, Algeria, Tunisia and Libya.

[2] Levantine covers the dialects spoken in Lebanon, Syria, Palestine and Jordan.

[3] Recently, two parallel Arabic translations were created for 12,000 sentences from the European Parliamentary proceedings, but both are in MSA (Habash et al., 2017).

The DARPA BOLT (Broad Operational Language Translation) project sponsored the creation of a large number of resources,[4] including a sizeable data set of DA sentences paired with their English translations. This data set consists of 2.2 million words of Egyptian and 1.5 million words of Levantine which were harvested from SMS messages and online sources like weblogs before being translated.

As for monolingual corpora, Zaidan and Callison-Burch (2011)'s Arabic Online Commentary (AOC) corpus contains 52 million words of mixed MSA and DA from news articles and readers' comments. Cotterell and Callison-Burch (2014) add modest amounts of Twitter data to this corpus, though we find the domain difference harmful for language modeling and drop it in our experiments. Khalifa et al. (2016)'s GUMAR corpus contains over 100 million words of Gulf Arabic and a smattering of other dialects, all taken from internet novels, a genre of long conversational novels shared anonymously on online forums popular among female teenagers. Other monolingual DA corpora like Tunisiya (McNeil and Faiza, 2011), the Curras corpus of Palestinian-Levantine (Jarrar et al., 2014), and those corpora presented in Al-Shargi et al. (2016), focus on different dialects or are too small to be relevant to Egyptian-to-Levantine MT.

As for DA-to-DA data, Bouamor et al. (2014) present the first corpus with 2,000 7-way parallel sentences of Egyptian, Tunisian, three Levantine dialects (Syrian, Jordanian, Palestinian), MSA, and English, all translated from Egyptian sentences harvested from the web. The authors concede that many Levantine sentences seem to be influenced by the Egyptian, likely because translators were primed with Egyptian expressions they might understand, but would not produce naturally. The same concern applies to the 6,400 sentence, 6-way parallel PADIC corpus used in Meftouh et al. (2015), as all translations were derived from DA or MSA. When developing the 12,000 sentence multi-dialectal corpus used in our experiments, we avoided such priming effects by asking translators to produce translations starting from English sentences taken from the Basic Travel Expressions Corpus (BTEC) (Takezawa et al., 2002).

Other relevant resources include AVIA,[5] a small but rich multi-dialect reference grammar with contextual examples, and Tharwa (Diab et al., 2014), a 4-way English, MSA, Egyptian, Levantine lexicon with rich linguistic annotation.

2.3 Pivot Machine Translation

Pivoting is an MT technique used to combat data sparsity when more source-to-pivot and pivot-to-target data is available than source-to-target parallel data (Muraki, 1987; Hajič et al., 2004; Wu and Wang, 2007; Habash and Hu, 2009). In this work, we use a specific form of pivoting: phrase pivoting. This involves aligning source-to-pivot and pivot-to-target data, extracting pairs of phrases into two phrase tables, then combining them into a single source-to-target phrase table based on shared pivot phrases (Utiyama and Isahara, 2007).

Our work is similar to El Kholy et al. (2013), who use English to translate from Persian to Arabic via phrase pivoting. They introduce connectivity strength constraints to weight learned Persian-to-Arabic phrase-pairs in the table by considering how well each pair can be aligned through an English pivot phrase (discussed further in Sections 4 and 5). In follow-up work, El Kholy and Habash (2015) add morphological constraints for translating related, morphologically rich languages Arabic and Hebrew, via morphologically-poor English. These constraints help preserve fine grained morphological distinctions like gender agreement which cannot otherwise be accurately translated via a morphologically poor pivot that does not make such distinctions, i.e., English.

[4] Pointers to the Linguistic Data Consortium's BOLT resources can be found here: `https://www.ldc.upenn.edu/collaborations/current-projects/bolt`.

[5] `http://www.umventures.org/technologies/arabic-variant-identification-aid-avia`

3 Data Preparation

All data used in our experiments comes from sources mentioned in Section 2.2. As displayed in Table 1, we split our 12,000 sentence BTEC parallel corpus into training, tuning, dev, and blind test sets, which are constant across all experiments. Also, in some experiments, we use additional monolingual and pivot data from AOC and the BOLT corpus, respectively.

Data Set	Dialect	Description	Size
BTEC-train	Egy–Lev	Parallel	8,000
BTEC-tune	Egy–Lev	Parallel	500
BTEC-dev	Egy–Lev	Parallel	1,500
BTEC-test	Egy–Lev	Parallel	2,000
BOLT-egy	Egy–Eng	Pivot	410,000
BOLT-lev	Eng–Lev	Pivot	180,000
AOC-egy	Egy	Monolingual	9,000
AOC-lev	Lev	Monolingual	5,000

Table 1: Data used in all experiments. Size reported in number of sentences.

Similar to MSA, DA is morphologically and syntactically rich, posing several challenges for MT systems. To be able to leverage morpho-syntactic features, we ran our Egyptian and Levantine data through MADAMIRA (Pasha et al., 2014), an Arabic morphological analyzer and disambiguator trained for MSA (MADAMIRA-MSA) and Egyptian (MADAMIRA-EGY). Unfortunately, the Levantine version of MADAMIRA is still under development (Eskander et al., 2016), so we use MADAMIRA-EGY to process both our Egyptian and Levantine corpora. Jarrar et al. (2014) and Khalifa et al. (2016) show that using MADAMIRA-EGY to process non-Egyptian DA data yields better results than MADAMIRA-MSA. To minimize the analyzer's bias towards Egyptian when processing Levantine data, we do not allow it to make orthographic changes. This limits the effects of misanalyzing many Levantine words, such as هالحظ *hAlHĎ* 'this luck', which can be incorrectly Epgyptian*ized* as حألحظ *HÂlHĎ* – the Egyptian future particle +ح *H+* together with an MSA verb ألحظ *ÂlHĎ* 'I perceive'.[6]

A number of tokenization and segmentation schemes are available for Arabic (Habash, 2010). Some separate only punctuation and digits. Others, such as ATB and D3, separate different sets of clitics from the base word. Whereas D3 segments all clitics, ATB leaves attached the definite article, ال*Al*. The optimal segmentation for our task is D3 (Sadat and Habash, 2006), as the aggressive tokenization mitigates for data sparsity. Typically, these tokenization schemes involve orthographic rewrite rules to ensure that the base word matches its non-cliticized form to minimize sparsity (El Kholy and Habash, 2012). Such rules depend on the morphological template of the word and the clitics attached to it. For a word such as حيكتبوها *HyktbwhA* 'and they will write it', the basic D3 tokenization is *H+ yktbwA +hA*. The extra *A* is added to the base word to minimize sparsity as this is how it would appear if no suffix had been appended.[7]

Since we do not have ideal tools for processing (tokenizing and detokenizing) Levantine, we opt for a stricter surface-word-oriented segmentation that guarantees recovering the form by simple concatenation when detokenizing. Thus, for حيكتبوها *HyktbwhA* 'and they will write it', the desired D3 segmentation is *H+ yktbw +hA*. This may increase data sparsity slightly, but more importantly, as mentioned previously, this limits the extent to which words can be

[6] Arabic transliteration is presented in the Habash-Soudi-Buckwalter scheme (Habash et al., 2007).

[7] In all of the work presented in this paper we apply ا/ى *Alif/Ya* normalization (El Kholy and Habash, 2012).

			Required Data		
Model	**BLEU**	**Out-of-vocabulary**	**Parallel**	**Monolingual**	**Pivot**
NO-TRANSLATION	6.48	N/A			
DIRECT	15.44	4.6	X		
SYNTHETIC	16.75	0.8	X	X	
PHRASE PIVOT	6.77	1.4			X
DIR+PP	**17.41**	0.9	X		X
SYNTHETIC-DIR+PP	16.81	0.8	X	X	X

Table 2: Baseline BLEU scores given different requirements: parallel, monolingual, or pivot data. Out-of-vocabulary rates are presented as percentages for each model.

misanalyzed or overly Egyptian*ized*. To achieve this, we extend a DA morphological database with suffix and prefix segmentations, adding a wrapper on top of MADAMIRA to generate the proper segmentation for each analysis. The database extension is automatic and the segmentation is deterministic, following D3 segmentation rules. This allows us to (i) apply this extension to other databases in other dialects that follow the structure of the MADAMIRA database, and (ii) expand our application to dialects that do not have any available analyzers yet.

4 Baseline Models

We use the phrase-based statistical MT platform, Moses (Koehn et al., 2007) to build multiple Egyptian-to-Levantine MT systems: one that only trains on parallel data, another that fabricates pseudo-parallel training data from additional monolingual data, and a third model utilizing pivot data through English. While neural MT has been successfully applied to MSA (Almahairi et al., 2016), we opt for statistical MT as data sparsity and other factors render neural techniques impossible for DA (Zhang et al., 2016). Luong and Manning (2015)'s English-to-Vietnamese neural MT system, for instance, leverages 10 times more parallel data then we use in our experiments, yet still fails to outperform a statistical baseline. Furthermore, their training and testing data is from a single domain with standardized spelling, i.e., limited token:type ratio, which Farajian et al. (2017) suggest should greatly facilitate neural MT performance. Given our sparsity of DA data and lack of spelling conventions, we can neither rely on homogeneous training/testing domains nor low token:type ratios and must resort to statistical MT.

We evaluate the output of our MT systems via BLEU scores (Papineni et al., 2002), comparing them to a single reference in detokenized space. NO-TRANSLATION, scoring 6.48, compares the original, unchanged Egyptian input to the Levantine reference. The results as well as data requirements are reported in Table 2.

4.1 The Direct Model

The most basic statistical system, the DIRECT model can be extended to any dialect pair with parallel data. It is trained only on our BTEC parallel corpus, with some additional monolingual data for language modeling. This model leverages a 2.4 million token 5-gram language model trained using KenLM (Heafield, 2011), consisting of Levantine data from the AOC corpus, BOLT, and BTEC.

Following El Kholy and Habash (2015), we perform word alignment using the grow-diag-final algorithm (Och and Ney, 2003) and we restrict the maximum length of extracted phrases to 8 tokens. Our D3 tokenization is slightly more aggressive than El Kholy and Habash (2015) who use ATB, so we experimented with marginal increases in the maximum allowable phrase length but found them to have no significant effects on performance.

As shown in Table 2, this basic model greatly outperforms the NO-TRANSLATION baseline at 15.44 BLEU, but suffers from a high rate of out-of-vocabulary (OOV) words given that it is only trained on a small amount of parallel data. Furthermore, the model seems to learn noisy weights for many of the phrase pairs it extracts due to the infrequency with which they are encountered during training.

4.2 The Synthetic Model

Inspired by Schwenk and Senellart (2009), we use additional monolingual data to build a SYNTHETIC MT system. First, we use the DIRECT model to translate all of the BOLT Egyptian data to Levantine. Then we build an inverse model identical to the DIRECT model, but from Levantine to Egyptian, and use it to translate the BOLT Levantine data into Egyptian. Finally, we learn a new phrase table from our newly generated parallel corpus consisting of the original 8,000 training sentences, 410,000 BOLT Egyptian-to-generated-Levantine sentences, and 180,000 BOLT Levantine-to-generated-Egyptian sentences.

While Schwenk and Senellart (2009) implement this technique in a slightly different manner for the purpose of domain adaptation, we use it to reduce noise in the phrase table. Due to sparsity of parallel data, the DIRECT model is hard pressed to distinguish good low frequency phrase pairs from bad ones. Adding synthetic data to the model enables it to learn better alignments for low frequency phrase pairs by getting exposure to a variety of different contexts in which such phrases can occur. This system significantly improves over the DIRECT model, scoring 16.75 BLEU, representing our best solution for DA-to-DA MT that does not require pivot data.

4.3 The Phrase Pivot Model

Following El Kholy et al. (2013), we use the BOLT data to phrase pivot through English. Phrase pivoting drastically increases vocabulary coverage; however, it also produces a phrase table with many poorly connected phrase pairs as well as phrase pairs which erroneously translate morpho-syntactic features that cannot be conveyed through morphologically-poor English. The PHRASE PIVOT model addresses the poor connectivity issue by adding El Kholy et al. (2013)'s connectivity strength constraints. These identify how many Egyptian and Levantine tokens in a given Egyptian-to-Levantine-via-English phrase pair can be aligned to each other via corresponding alignments to the same English token.

For example, the noisy Egyptian-to-Levantine phrase pair in Figure 1, would receive a connectivity score of 0.75 from the Egyptian side because 3 of the 4 alignments—those to 'wants', 'to', and 'go'—connect through the English pivot phrase to a Levantine token on the other side. The connectivity score from the Levantine side would be 0.6 because 3 of the 5 Levantine alignments connect all the way through. *hlq* does not count towards the 3 connections because while it connects to the English token 'now', no Egyptian token connects to 'now' from the other side. This example also exhibits the issue that will be addressed in Section 5, that morpho-syntactic properties are not accurately conveyed through morphologically deprived English, as *ʕAyz* and *bd* connect through 'want', though *ʕAyz* implies a masculine subject whereas the suffix of *bd*, *hA*, entails that the subject of the Levantine sentence is in fact third-person feminine.

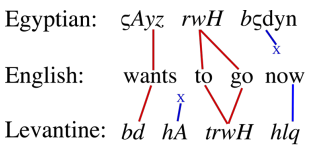


Figure 1: Identifying the connections between an Egyptian phrase and Levantine phrase which were both independently (and noisily) mapped to the same English phrase during pivoting.

This PHRASE PIVOT model can be extended to any DA pair with pivot data, but only marginally outperforms the NO-TRANSLATION baseline at 6.77 BLEU. However, used in conjunction with the DIRECT model, the Direct + Phrase Pivot (DIR+PP) model increases OOV coverage, boosting performance to 17.41 BLEU, almost 2 full BLEU points over the DIRECT baseline. We also re-ran the SYNTHETIC model using DIR+PP to fabricate parallel data instead of DIRECT, however, this did not improve performance. It is possible that the types of DIRECT model errors which are corrected by DIR+PP versus those corrected by SYNTHETIC are similar. Thus, when training on fabricated parallel data, the SYNTHETIC-DIR+PP model may reinforce its own errors more than learn to fix them.

5 Leveraging Morphology and Syntax

The best baseline system, DIR+PP still fails to adequately handle Arabic's rich morphology and syntax, as illustrated by Figure 2, where part-of-speech (POS) is not preserved in the output. A minimally different correct version of the example in Figure 2 would simply replace verbal third-person singular بيفتح *byftH* 'he opens', with the nominal form فتح في *fy ftH* 'in opening'.

Source:				انا عندي مشكلة [[في فتح]] الباب	
	AnA	*ʕndy*	*mšklħ*	[[*fy ftH*]]	*AlbAb*
	I	to-me	problem	[[in opening.N]]	the-door
Output:				انا عندي مشكلة [[بيفتح]] الباب	
	AnA	*ʕndy*	*mšklħ*	[[*byftH*]]	*AlbAb*
	I	to-me	problem	[[with-opens.3MS]]	the-door
Reference:	I'm having trouble opening the door				

Figure 2: Example DIR+PP error where the output does not preserve the POS of the source.

Because Arabic verbs convey person in much finer granularity than do English verbs, which only inflect for third-person singular forms in present tense, many Arabic verb inflections in the source-to-pivot and pivot-to-target phrase tables will be aligned to the same morphologically deprived English verb, e.g., 'opening'. Thus, when the phrase tables are combined via shared English phrases, any given inflected Egyptian verb can be mapped to a large number of Levantine inflections, which, mostly, will not share the same morpho-syntactic properties. In this case, because 'opening', like many 'ing'-suffixed forms in English, can be nominal or verbal, it is not just inflectional morphology that is confused but derivational morphology, as the POS is misinterpreted.

5.1 Addressing Resource Limitations

El Kholy and Habash (2015) use AMEANA (El Kholy and Habash, 2011), an automatic error analysis tool, to determine that definiteness, gender, and number are the features that most frequently contribute to such errors in Hebrew-to-MSA MT. In this work, we were not able to use AMEANA, as it relies on accurate morphological analyses that we cannot produce automatically for Levantine. Furthermore, even if we knew what were the most problematic features for translating Egyptian to Levantine, we might not be able to leverage them, as there is non-trivial noise and Egyptian bias in how the analyses were generated (Section 3).

Our approach focuses instead on identifying features that: (i) MADAMIRA-EGY can recognize relatively accurately (ii) tend to be consistently translated from Egyptian to Levantine. For instance, second-person and third-person verbal forms are frequently orthographically ambiguous in Arabic, making person challenging for our analyzer to correctly identify. Thus,

adding a constraint to the model promoting consistent translation of the person feature value would be useless because we are not likely to know the correct property of person in the first place. Furthermore, if the possible values of a given feature can be translated freely, modeling that feature will be similarly useless. This is often the case with tense, as بشوفك *bšwfk* 'see you' in Egyptian could conceivably be translated as رح شوفك *rH šwfk* in Levantine, changing the value from progressive (realized by the cliticized particle ب *b*) to future tense (realized by the particle رح *rH*).

Without gold, morphologically annotated data in Levantine, we cannot independently measure either our ability to identify morpho-syntactic feature values correctly, or the consistency with which they should be translated. However, we can approximate both jointly. Assuming that Egyptian feature values should correspond to the same feature values on the Levantine side, we measure, for each morpho-syntactic feature, how frequently the realized property on the Egyptian side is aligned to the same property on the Levantine side.

As shown in Table 3, definiteness, gender, number, and POS are the only features which map consistently across aligned tokens in more than 50% of their occurrences throughout the BTEC training set. This suggests that they both can be recognized accurately and are consistently preserved in human translation. Even so, the fact that none of these features map consistently over 80% of the time, suggests that modeling such features will be noisy.

Definiteness	**Number**	**Gender**	**POS**	**Aspect**	**Person**
75	75	62	56	32	29

Table 3: Percentage rates over all training set token alignments at which the feature's values were preserved from Egyptian to Levantine.

5.2 Computing Constraint Scores

Similar to El Kholy and Habash (2015), we design morpho-syntactic constraints by calculating probability distributions from Egyptian to Levantine and vice versa. These reflect how likely each morpho-syntactic property set on one side is to be aligned to each morpho-syntactic property set on the other, based on how often such alignments occured in the training data. Properties sets are defined as the conjunction of values for all morpho-syntactic features under consideration, which for us, include the four most "consistent" features as identified in Table 3: definiteness, number, gender—which were used by El Kholy and Habash (2015)—and also POS (thus, masculine-singular-verb and definite-feminine-noun are property sets). Unaligned tokens are considered to be aligned to a null token on the opposite side, and thus are mapped to an empty property set. We use these probability distributions to add two constraint scores to every phrase pair in the phrase pivot table, one calculated from Egyptian to Levantine and the other, Levantine to Egyptian, as defined in Equations 1 and 2.

$$W_s = \frac{1}{A} \sum_{\forall (i,j) \in a} P(MLE(i)|MLE(j)) \tag{1}$$

$$W_t = \frac{1}{B} \sum_{\forall (i,j) \in b} P(MLE(j)|MLE(i)) \tag{2}$$

We calculate W_s by summing over every alignment a from source token i to target token j (if i is unaligned, j is the null token), the probability of i's property set given j's. While El Kholy and Habash (2015) normalize this sum by the quantity of source tokens, we normalize by the total number of alignments A. Otherwise, many-to-one and one-to-many alignments would bias the scores and enable some to exceed one, making them impossible to interpret as probabilities. The property sets of i and j are determined from the set of all possible property sets for that type (defined as the list of all unique analyses it received over every occurrence in BOLT, AOC, and the BTEC training data) via maximum likelihood estimation, MLE, so as to maximize the likelihood of the source property set given the target property set.

This entails that individual tokens' property sets can be analyzed differenlty from the source side than from the target side. Also, sequences of MLE property sets over multiple tokens on a single side can be syntactically infeasible, e.g., containing 5 consecutive verbs. Thus, we experimented with additional constraints, requiring (i) aligned MLE property sets to have been aligned at least once in the training set (ii) syntactic feasibility on source and target sides independently (iii) alignment of the sequence of property sets on the source side to that on the target side to have appeared at least once in the training set. However, none of these experiments boosted performance, suggesting that MLE inconsistency is not a problematic issue.

W_t is calculated equivalently to W_s from the target side. Adding these constraint weights to each phrase pair in the phrase pivot table, we re-tune the DIR+PP system, re-test, and obtain a statistically significant improvement with a score of 18.03 BLEU on the development set.

We then evaluate the DIRECT, DIR+PP, and Direct + Phrase Pivot with Morpho-syntactic Features (DIR+PP+MORPH) systems on the 2,000 sentence blind test set from our BTEC corpus. The results in Table 4 confirm the utility of our added constraints, as each successive model significantly improves over the last, as in the development set.

Model	Dev	Dev OOV	Test	Test OOV
NO-TRANSLATION	6.48	N/A	6.45	N/A
DIRECT	15.44	4.6	14.61	5.4
DIR+PP	17.41	0.9	16.69	1.0
DIR+PP+MORPH	**18.03**	0.9	**17.48**	1.0

Table 4: Comparing BLEU scores of systems with and without morpho-syntactic features on development and blind test sets. Out-of-vocabulary rates are reported as percentages.

6 Error Analysis

We analyzed the output of the DIR+PP and DIR+PP+MORPH models on 100 development set sentences to investigate the effects of morpho-syntactic features and to identify issues for future work. Table 5 reveals a stark contrast—about a 30% gap in both models—between the quantity of output tokens matching a reference token letter-for-letter (row 3), and the quantity of output tokens manually judged to be acceptable (row 2). Approximately 10% of that 30% gap is due to lack of spelling standards in DA (row 4). Habash et al. (2012a) developed a Conventional Orthography for Dialectal Arabic (CODA) to standardize spelling while preprocessing DA and a prototype system can CODA*fy* Egyptian (Eskander et al., 2013), though no such system is yet available for Levantine. Once developed, we expect such a system to improve MT quality not only by imposing consistent output, but also by reducing sparsity in all translation and language models during training.

The remaining 90% of the 30% gap between exactly matched tokens and tokens judged

to be correct can be approximately split into thirds. One third cannot be directly linked to any reference token (row 7), e.g., tokens in paraphrasal or idiomatic constructions. Another third is tokens which can be linked to a reference token, but take a different root (row 5). The final third are inflectional or derivational variants of the corresponding reference token (row 6). The fact that so many inflectional/derivational variants are judged correct demonstrates that morpho-syntactic modeling is necessarily noisy as property sets are frequently not preserved, even in acceptable translations. On the other hand, the success of DIR+PP+MORPH suggests that some features' properties tend to be preserved through translation or at least altered predictably, as otherwise, the system would not benefit from modeling them.

		DIR+PP	DIR+PP MORPH
1	**Words**	665	670
2	**Words Judged Correct**	**569 (85.6)**	**594 (88.7)**
3	Exact Match	377 (56.7)	382 (57.0)
4	CODA Variant	23 (3.5)	25 (3.7)
5	Different Root	47 (7.1)	51 (7.6)
6	Different Properties	61 (9.2)	65 (9.8)
7	Otherwise Different	61 (9.2)	71 (10.7)
8	**Words Judged Incorrect**	**96 (14.4)**	**76 (11.3)**
9	Morpho-syntactic Properties	49 (7.4)	41 (6.1)
10	Other Problems	47 (7.1)	35 (5.2)
11	**Properties**: Modeled	33 (5.0)	26 (3.9)
12	Not Modeled	16 (2.4)	15 (2.2)
13	**Other**: Wrong Word Sense	18 (2.7)	17 (2.5)
14	Apparent Phrasal Issue	13 (2.0)	7 (1.0)
15	Unclear Reason	13 (2.0)	7 (1.0)
16	OOV	2 (0.3)	3 (0.4)
17	Copies Egyptian	1 (0.2)	1 (0.1)
18	**Word Error Reduction**	**N/A**	**(20.2)**
19	**Sentences**	100	100
20	Correct Sentences	48	55
21	Sentence Error Reduction	N/A	(13.5)

Table 5: Comprehensive manual error analysis of 100 sentences from the development set. Values within parentheses are percentages.

6.1 Direct Effects of Added Features

The second major insight of the error analysis is that the error reduction from adding morpho-syntactic constraints is far more significant (20.2%, row 18), than the improvement registered by the automatic BLEU scores. Example sentences illustrating some of these improvements are contained in Figure 3. For 7.4% of the tokens DIR+PP outputs, its only mistake is mis-representing one or more morpho-syntactic property (row 9). Comparing that to 6.1% for DIR+PP+MORPH (row 9), the new model makes a 17% error reduction in the area it is designed to improve. Furthermore, essentially all of this improvement takes place in sentences where DIR+PP+MORPH corrects a mistake involving a feature we model: definiteness, gender, number, or POS. For example, the definite article is correctly added to the word الحقيقة *AlHqyqħ* 'the truth' in Figure 3a.

	ENGLISH:	Actually, inside it [...] Do you mind if I take it out?							
	REFERENCE:	بالحقيقة في بالبو [...] عندك مانع ازا شلتو؟							
		bAlHqyqħ	*fy*	*bAlbw*	[...]	*ʕndk*	*mAnʕ*	*AzA*	*šltw?*
		with-the-truth	exists	in-heart-its	[...]	to-you	problem	if	take-1SPast-it
(a)	**DIR+PP:**	حقيقة* في بالبا [...] عندك مانع اذا طلعو ؟*							
		*Hqyqħ**	*fy*	*bAlbA*	[...]	*ʕndk*	*mAnʕ*	*AðA*	*Tlʕw?**
		truth*	exists	in-heart-its	[...]	to-you	problem	if	remove.3SPast-it*
	DIR+PP+MORPH:	الحقيقة ، هوي بقلبو [...] عندك مانع اذا شلتو؟							
		AlHqyqħ	*hwy*	*bqlbw*	[...]	*ʕndk*	*mAnʕ*	*AðA*	*šltw?*
		the-truth	it	in-heart-its	[...]	to-you	problem	if	take.1SPast-it?

	ENGLISH:	I'll bring one right away				
	REFERENCE:	رح جيب واحد هلا				
		rH	*jyb*	*wAHd*	*hlA*	
		will	bring.1S	one	now	
(b)	**DIR+PP:**	بالجيبة* واحد هلق				
		*bAljybħ**	*wAHd*	*hlq*		
		with-the-pocket*	one	now		
	DIR+PP+MORPH:	رح جيب واحد هلق				
		rH	*jyb*	*wAHd*	*hlq*	
		will	bring.1S	one	now	

	ENGLISH:	What kind of fruit do you have?					
	REFERENCE:	اي نوع فواكي عندك؟					
		Ay	*nwʕ*	*fwAky*	*ʕndk?*		
		which	kind	fruit	at-you		
(c)	**DIR+PP:**	عندك اي نوع من الفاكهة شو؟*					
		ʕndk	*Ay*	*nwʕ*	*mn*	*AlfAkhħ*	*šw?**
		at-you	which	kind	from	the-fruit	what*
	DIR+PP+MORPH:	عندك اي نوع من الفاكهة؟					
		ʕndk	*Ay*	*nwʕ*	*mn*	*AlfAkhħ?*	
		at-you	which	kind	from	the-fruit	

Figure 3: Example translation errors (marked with *) corrected by adding morpho-syntactic constraints to the model.

6.2 Indirect Effects of Added Features

Surprisingly, most of the overall error reduction actually comes from mistakes other than misrepresentation of morpho-syntactic properties, as such mistakes decrease by 26%, from 7.1% to 5.2% (row 10). The morpho-syntactic constraints seem to teach the model about syntax at the phrase level, as sentences like Figure 3b are corrected, which were originally over-chunked into small phrases by DIR+PP. بالجيبة *bAljybħ* 'with the pocket' is likely a misanalysis of the source word هاجيب *hAjyb* 'I'll get' translated as a single-word phrase, as it would have made for an infrequent or non-existent bigram or trigram when combined with the following words in the output. The DIR+PP model likely had access to longer phrase pairs such as هاجيب واحد *hAjyb wAHd* 'I'll get one' mapping to رح جيب واحد *rH jyb wAHd*—the corresponding reference phrase—but likely did not select it because longer phrases are inherently less frequent,

i.e., noisier to model.

Morpho-syntactic constraints enable DIR+PP+MORPH to increasingly select longer, infrequent phrase pairs by distinguishing those that are morpho-syntactically feasible from competing shorter alternatives. Translating larger phrasal chunks leads to more fluent output by reducing opportunities to incorrectly chunk phrase boundaries. This is why much of the error reduction does not appear to be, superficially at least, related to morpho-syntactic features targeted by DIR+PP+MORPH.

Figure 3c exhibits another benefit of morpho-syntactic features, as DIR+PP+MORPH often corrects the insertion of gratuitous words, here, شو *šw* 'what'. Such features teach the model that certain POS's are less likely to align to the null token, even if the language model favors the sequence with the gratuitous token monolingually.

7 Conclusion and Future Work

In this work, we presented the second ever evaluated Arabic DA-to-DA MT effort. The subject has not previously received serious attention due to lack of naturally occurring parallel data, though DA is widely spoken and dialects are frequently mutually unintelligible, exhibiting comparable linguistic variation to the Romance languages. Our results suggest that modeling morphology and syntax can significantly improve DA-to-DA MT despite data sparsity. However, optimizing models under such circumstances requires careful consideration of the linguistic differences between dialects and careful tailoring and implementation of all available data and resources.

Given that many DA pairs may not have pivot data available, the most pressing future work is to develop a dialect-agnostic tokenizer and analyzer which does not suffer from the Egyptian bias that ours does. This will reduce data sparsity regardless of the nature of the low-resourced MT settings for any DA pair, and it will enable better morpho-syntactic modeling.

Additionally, improving on the dialect identification work of Diab et al. (2010), Zaidan and Callison-Burch (2011), and Elfardy and Diab (2013) will enable us to collect more monolingual data. This data is not only useful for language modeling but can also be mined for comparable sentences to augment the parallel training set. The process typically involves using metadata (Resnik and Smith, 2003) and seed data (Munteanu and Marcu, 2005) to identify pairs of related sentences or phrases in the source and target languages (Cettolo et al., 2010; Max et al., 2012). These are then iteratively classified via expectation maximization with phrases identified as parallel being added to the seed data (Dong et al., 2015). Models trained thusly produce noisy phrase pairs, often imperfectly modeling morpho-syntactic property sets. Thus, the same morpho-syntactic constraints that improved DIR+PP+MORPH can be adapted to improve MT via comparable corpora.

8 Acknowledgments

This publication was made possible by grant NPRP 7-290-1-047 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors. The first author was funded by the Boren Fellowship program.

References

Al-Shargi, F., Kaplan, A., Eskander, R., Habash, N., and Rambow, O. (2016). Morphologically annotated corpora and morphological analyzers for Moroccan and Sanaani Yemeni Arabic. In *10th Language Resources and Evaluation Conference (LREC 2016)*.

Almahairi, A., Cho, K., Habash, N., and Courville, A. (2016). First result on Arabic neural machine translation. *arXiv preprint arXiv:1606.02680*.

Armentano-Oller, C., Carrasco, R. C., Corbí-Bellot, A. M., Forcada, M. L., Ginestí-Rosell, M., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., Sánchez-Martínez, F., and Scalco, M. A. (2006). Open-source Portuguese–Spanish machine translation. In *International Workshop on Computational Processing of the Portuguese Language*, pages 50–59.

Bouamor, H., Habash, N., and Oflazer, K. (2014). A multidialectal parallel corpus of Arabic. In *LREC*, pages 1240–1245.

Cettolo, M., Federico, M., and Bertoldi, N. (2010). Mining parallel fragments from comparable texts. In *IWSLT*, pages 227–234.

Chiang, D., Diab, M., Habash, N., Rambow, O., and Shareef, S. (2006). Parsing Arabic Dialects. In *Proceedings of EACL*, Trento, Italy. EACL.

Corbí Bellot, A. M., Forcada, M. L., Ortiz Rojas, S., Pérez-Ortiz, J. A., Ramírez Sánchez, G., Sánchez-Martínez, F., Alegría Loinaz, I., Mayor Martínez, A., Sarasola Gabiola, K., et al. (2005). An open-source shallow-transfer machine translation engine for the Romance languages of Spain. *Proceedings of the 10th Conference of the European Association for Machine Translation*, pages 79–86.

Cotterell, R. and Callison-Burch, C. (2014). A multi-dialect, multi-genre corpus of informal written Arabic. In *LREC*, pages 241–245.

Diab, M., Habash, N., Rambow, O., Altantawy, M., and Benajiba, Y. (2010). Colaba: Arabic dialect annotation and processing. In *LREC Workshop on Semitic Language Processing*, pages 66–74.

Diab, Mona T and Al-Badrashiny, Mohamed and Aminian, Maryam and Attia, Mohammed and Elfardy, Heba and Habash, Nizar and Hawwari, Abdelati and Salloum, Wael and Dasigi, Pradeep and Eskander, Ramy. (2014). Tharwa: A Large Scale Dialectal Arabic-Standard Arabic-English Lexicon. In *LREC*, pages 3782–3789.

Dong, M., Liu, Y., Luan, H.-B., Sun, M., Izuha, T., and Zhang, D. (2015). Iterative learning of parallel lexicons and phrases from non-parallel corpora. In *IJCAI*, pages 1250–1256.

Durrani, N., Al-Onaizan, Y., and Ittycheriah, A. (2014). Improving Egyptian-to-English SMT by mapping Egyptian into MSA. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 271–282.

El Kholy, A. and Habash, N. (2011). Automatic error analysis for morphologically rich languages. In *MT Summit XIII*.

El Kholy, A. and Habash, N. (2012). Orthographic and morphological processing for English–Arabic statistical machine translation. *Machine Translation*, 26(1-2):25–45.

El Kholy, A. and Habash, N. (2015). Morphological Constraints for Phrase Pivot Statistical Machine Translation. In *Machine Translation Summit*, Miami.

El Kholy, A., Habash, N., Leusch, G., Matusov, E., and Sawaf, H. (2013). Language independent connectivity strength features for phrase pivot statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 412–418.

Elfardy, H. and Diab, M. (2013). Sentence Level Dialect Identification in Arabic. In *Proceedings of the Association for Computational Linguistics*, pages 456–461, Sofia, Bulgaria.

Eskander, R., Habash, N., Rambow, O., and Pasha, A. (2016). Creating resources for Dialectal Arabic from a single annotation: A case study on Egyptian and Levantine. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3455–3465, Osaka, Japan.

Eskander, R., Habash, N., Rambow, O., and Tomeh, N. (2013). Processing Spontaneous Orthography. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, GA.

Farajian, M. A., Turchi, M., Negri, M., Bertoldi, N., and Federico, M. (2017). Neural vs. phrase-based machine translation in a multi-domain scenario. *EACL 2017*, page 280.

Habash, N., Diab, M., and Rambow, O. (2012a). Conventional Orthography for Dialectal Arabic. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 711–718, Istanbul, Turkey.

Habash, N., Eskander, R., and Hawwari, A. (2012b). A Morphological Analyzer for Egyptian Arabic. In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, pages 1–9, Montréal, Canada.

Habash, N. and Hu, J. (2009). Improving Arabic-Chinese Statistical Machine Translation using English as Pivot Language. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 173–181, Athens, Greece.

Habash, N., Soudi, A., and Buckwalter, T. (2007). On Arabic Transliteration. In van den Bosch, A. and Soudi, A., editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Habash, N., Zalmout, N., Taji, D., Hoang, H., and Alzate, M. (2017). A parallel corpus for evaluating machine translation between arabic and european languages. *EACL 2017*, page 235.

Habash, N. Y. (2010). *Introduction to Arabic natural language processing*, volume 3. Morgan & Claypool Publishers.

Hajič, J., Smrž, O., Zemánek, P., Šnaidauf, J., and Beška, E. (2004). Prague Arabic Dependency Treebank: Development in data and tools. In *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, pages 110–117, Cairo, Egypt.

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197.

Jarrar, M., Habash, N., Akra, D., and Zalmout, N. (2014). Building a corpus for Palestinian Arabic: a preliminary study. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 18–27.

Khalifa, S., Habash, N., Abdulrahim, D., and Hassan, S. (2016). A large scale corpus of Gulf Arabic. *arXiv preprint arXiv:1609.02960*.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Koehn, P., Birch, A., and Steinberger, R. (2009). 462 Machine Translation Systems for Europe. In *Proceedings of MT Summit*, Ottawa, Canada.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180.

Luong, M.-T. and Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*.

Maamouri, M., Bies, A., Kulick, S., Ciul, M., Habash, N., and Eskander, R. (2014). Developing an Egyptian Arabic treebank: Impact of dialectal morphology on annotation and tool development. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Max, A., Bouamor, H., and Vilnat, A. (2012). Generalizing sub-sentential paraphrase acquisition across original signal type of text pairs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 721–731.

McNeil, K. and Faiza, M. (2011). Tunisian Arabic corpus: Creating a written corpus of an unwritten language. In *Workshop on Arabic Corpus Linguistics (WACL)*.

Meftouh, K., Harrat, S., Jamoussi, S., Abbas, M., and Smaili, K. (2015). Machine translation experiments on PADIC: A parallel Arabic dialect corpus. In *The 29th Pacific Asia conference on language, information and computation*.

Munteanu, D. S. and Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Muraki, K. (1987). PIVOT: Two-phase machine translation system. In *MT Summit Manuscripts and Program*, pages 81–83.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318.

Pasha, A., Al-Badrashiny, M., Kholy, A. E., Eskander, R., Diab, M., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *Proceedings of LREC*, Reykjavik, Iceland.

Resnik, P. and Smith, N. A. (2003). The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

Sadat, F. and Habash, N. (2006). Combination of Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1–8.

Sajjad, H., Darwish, K., and Belinkov, Y. (2013). Translating dialectal Arabic to English. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–6, Sofia, Bulgaria.

Salloum, W. and Habash, N. (2011). Dialectal to standard arabic paraphrasing to improve arabic-english statistical machine translation. In *Proceedings of the first workshop on algorithms and resources for modelling of dialects and language varieties*, pages 10–21.

Salloum, W. and Habash, N. (2012). Elissa: A Dialectal to Standard Arabic Machine Translation System. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012): Demonstration Papers*, pages 385–392, Mumbai, India.

Sawaf, H. (2010). Arabic dialect handling in hybrid machine translation. In *Proceedings of AMTA*, Denver, Colorado.

Schwenk, H. and Senellart, J. (2009). Translation model adaptation for an Arabic/French news translation system by lightly-supervised training. In *MT Summit*.

Tachicart, R. and Bouzoubaa, K. (2014). A hybrid approach to translate Moroccan Arabic dialect. In *Intelligent systems: Theories and applications (sita-14), 2014 9th international conference on*, pages 1–5. IEEE.

Takezawa, T., Sumita, E., Sugaya, F., Yamamoto, H., and Yamamoto, S. (2002). Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *LREC*, pages 147–152.

Tyers, F. M., Font, H. A. i., Fronteddu, G., and Martín-Mor, A. (2017). Rule-based machine translation for the Italian-Sardinian language pair. In *The Prague Bulletin of Mathematical Linguistics No. 108*, pages 221–232.

Utiyama, M. and Isahara, H. (2007). A comparison of pivot methods for phrase-based statistical machine translation. In *HLT-NAACL*, pages 484–491.

Wu, H. and Wang, H. (2007). Pivot language approach for phrase-based statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 856–863, Prague, Czech Republic.

Zaidan, O. F. and Callison-Burch, C. (2011). The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content. In *Proceedings of the Association for Computational Linguistics*, Portland, Oregon, USA.

Zbib, R., Malchiodi, E., Devlin, J., Stallard, D., Matsoukas, S., Schwartz, R., Makhoul, J., Zaidan, O. F., and Callison-Burch, C. (2012). Machine translation of Arabic dialects. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 49–59.

Zhang, D., Kim, J., Crego, J. M., and Senellart, J. (2016). Boosting neural machine translation. *CoRR*, abs/1612.06138.

Elastic-substitution decoding for Hierarchical SMT: efficiency, richer search and double labels

Gideon Maillette de Buy Wenniger[1] gemdbw AT gmail.com
Khalil Sima'an[2] k.simaan AT uva.nl
Andy Way[1] andy.way AT adaptcentre.ie

[1]ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland
[2]Institute for Logic Language and Computation (ILLC), Faculty of Science, University of Amsterdam, Amsterdam, The Netherlands

Abstract

Elastic-substitution decoding (ESD), first introduced by Chiang (2010), can be important for obtaining good results when applying labels to enrich hierarchical statistical machine translation (SMT). However, an efficient implementation is essential for scalable application. We describe how to achieve this, contributing essential details that were missing in the original exposition. We compare ESD to strict matching and show its superiority for both reordering and syntactic labels. To overcome the sub-optimal performance due to the late evaluation of features marking label substitution types, we increase the diversity of the rules explored during cube pruning initialization with respect to labels their labels. This approach gives significant improvements over basic ESD and performs favorably compared to extending the search by increasing the cube pruning pop-limit. Finally, we look at combining multiple labels. The combination of reordering labels and target-side boundary-tags yields a significant improvement in terms of the word-order sensitive metrics Kendall reordering score and METEOR. This confirms our intuition that the combination of reordering labels and syntactic labels can yield improvements over either label by itself, despite increased sparsity.

1 Introduction

Elastic-substitution decoding (ESD) – also known as *soft label matching* or *soft-constraint decoding* – is an effective method to gain maximal benefit from the use of labels to enrich hierarchical phrase-based statistical machine translation (SMT), and was first introduced by Chiang (2010). This method removes many of the disadvantages of working with labeled grammars when labels are strictly enforced. We discuss the requirements and details of an efficient implementation in the first part of this paper, to benefit other researchers that want to apply ESD. In the second part of the paper we further strengthen the empirical evidence for the success of ESD. This is done by comparing strict and soft-labeled (ESD) systems for Chinese–English translation, using four different types of labels. Next, we describe how the results of ESD can be further improved for small label sets by diversifying the search, exploring all alternatively labeled versions of each rule source-side type during cube pruning initialization instead of only the single best one. This is compared against the more crude approach of just increasing the search space by increasing the cube pruning *pop-limit*. Finally, we explore the effect of combining multiple labels, either the two types of reordering labels or a reordering

label with a syntactic label. All source code for both ESD and labeled grammar extraction is made publicly available with this publication.[1]

2 Background and Related Work

Hierarchical phrase-based SMT (or hierarchical SMT for short) (Chiang, 2005) is the hierarchical generalization of phrase-based SMT (Koehn et al., 2003). It generalizes phrase-pairs into synchronous context-free grammar (SCFG) rules by adding variables to them. This yields a weighted SCFG (Aho and Ullman, 1969). The particular form of SCFGs used in this paper is called HIERO (Chiang, 2005), and allows only up to two nonterminals (variables) in the right-hand-side of rules. This gives the following four HIERO rule types:

$$X \rightarrow \langle \alpha,\ \delta \rangle \tag{1}$$

$$X \rightarrow \langle \alpha\ X_{\boxed{1}} \gamma,\ \delta\ X_{\boxed{1}} \eta \rangle \tag{2}$$

$$X \rightarrow \langle \alpha\ X_{\boxed{1}} \beta\ X_{\boxed{2}} \gamma\ ,\ \delta\ X_{\boxed{1}}\ \zeta\ X_{\boxed{2}} \eta\ \rangle \tag{3}$$

$$X \rightarrow \langle \alpha\ X_{\boxed{1}} \beta\ X_{\boxed{2}} \gamma\ ,\ \delta\ X_{\boxed{2}}\ \zeta\ X_{\boxed{1}} \eta\ \rangle \tag{4}$$

Here $\alpha, \beta, \gamma, \delta, \zeta, \eta$ are terminal sequences that can be empty, except for β, since HIERO prohibits rules with nonterminals that are adjacent on the source side. HIERO additionally requires all rules to have at least one pair of aligned words. These extra constraints are intended to reduce the amount of spurious ambiguity. Equation (1) corresponds to a normal phrase pair, (2) to a rule with one gap and (3) and (4) to the monotone and inverting rules, respectively.

In addition, HIERO has a special glue rule: (g1) $GOAL \rightarrow \langle GOAL_{\boxed{1}}\ X_{\boxed{2}},\ GOAL_{\boxed{1}}\ X_{\boxed{2}} \rangle$ as well as two special start/end rules: (g2) $GOAL \rightarrow \langle GOAL_{\boxed{1}}\ </s>\ ,\ GOAL_{\boxed{1}}\ </s> \rangle$ and (g3) $GOAL \rightarrow \langle <s>, <s> \rangle$, with $<s>$ and $</s>$ being the dedicated start/end symbols.

HIERO makes very strong independence assumptions, since it uses only one label "X" apart from the glue symbol *GOAL*, allowing any HIERO rule to substitute to any other rule. A lot of work has been done on relaxing these assumptions by labeling HIERO with labels derived from syntax (Zollmann and Venugopal, 2006; Almaghout et al., 2011), dependency information (Li et al., 2012), word classes such as POS-tags (Zollmann and Vogel, 2011), reordering information (Maillette de Buy Wenniger and Sima'an, 2014) and other types of information.

However, labeling with strict matching of labels splits the rules of HIERO into many alternatively labeled variants, increasing spurious ambiguity. Venugopal et al. (2009) introduced preference grammars as a way to avoid this increase and to relax the assumptions of decoding with strict matching. Every rule is equipped with label distributions instead of single labels, for both the left- and right-hand-side rule nonterminals. Using a dynamic programming approach, these label distributions are then multiplied during decoding, to approximate the probability over the full set of alternatively labeled derivations. Unlike preference grammars, ESD does not approximate selection of the most likely unlabeled derivation. However, in contrast it can learn to treat different substitutions such as NP → NPP differently from others such as NP → VP which the formalism of preference grammars cannot, as it lacks a learning component. This is a clear advantage for heuristically created labels such as syntax-augmented machine translation (SAMT) and others used in this paper.

[1] Source code URLs: *ESD*: https://github.com/gwenniger/joshua/commits/gideon/cubePruningFixForFuzzyMatching
Grammar extraction: https://bitbucket.org/gwenniger/labeled-translation https://bitbucket.org/teamwildtreechase/hatparsing

The work by Chiang (2010) on ESD, the foundation of the work in this paper, is discussed next.

3 Elastic-substitution decoding

ESD was introduced by Chiang (2010), who describes it as follows: "*... we allow any rule to substitute into any site, but let the model learn which substitutions are better than others.*"

With respect to decoding it is remarked that: "*The decoding algorithm then operates as in hierarchical phrase-based translation. The decoder has to store in each hypothesis the source and target root labels of the partial derivation, but these labels are used for calculating feature vectors only and not for checking well-formedness of derivations.*"

In summary ESD entails:

(A) Adapting the decoder to support soft-matching of labels, which means finding all matching rules while ignoring the labels.

(B) Adding *label-substitution features* that mark different types of substitutions: (i) matching and mismatching substitutions, and (ii) substitutions of particular types of labels to particular gaps to enable learning what type of substitutions are preferable.

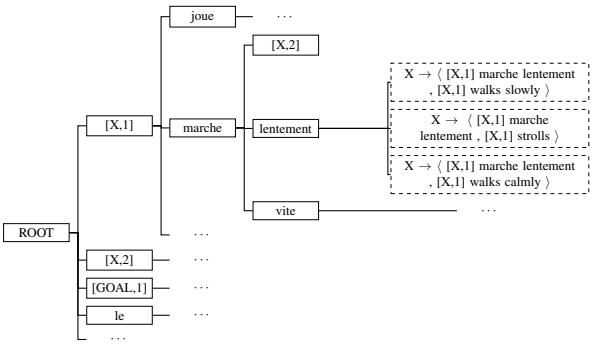
To enable computing the label-substitution features (B), the labels must be left present in the hypergraph (the packed hypotheses) computed by the decoder.

3.1 ESD: a naive implementation

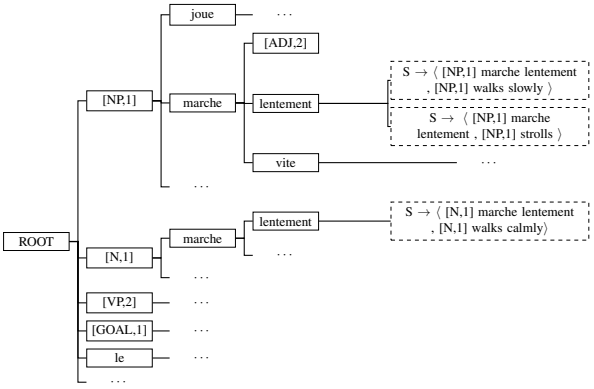
With strict matching, the inner loop of the decoder finds all matching rules r_{match} for an input word span $s_{<i,j>} = w_i \dots w_j$. For the rule right-hand-side $rhs = RHS(r_{match})$, given the ordered words $w_k \in rhs$ and nonterminals $nt_l \in rhs$, w_k must match the corresponding word in $s_{<i,j>}$ and nt_l must match the label of a corresponding chart span that was previously covered by the decoder (so-called "rule gap"); both must be matched in accordance with the input order. Adding ESD to this process, a naive implementation explicitly matches all best alternatively labeled rule variants for an (unlabeled) source-rule type, to all alternatively labeled gaps. This naive implementation is, however, computationally expensive. For rules with up to two gaps the number of source rule variants can increase quadratically with the size of the label set N, and analogously the same holds for the two substituted-to gaps of these rules. Hence this naive approach gives an increase in computational complexity of $O(N^4)$.

3.2 How is ESD implemented efficiently?

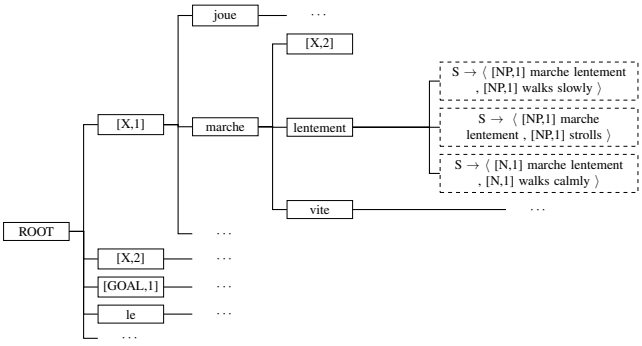
During normal (strict) decoding, matching rules are found through lookup in a dedicated rule indexing data-structure called a *trie* (De La Briandais, 1959). An efficient ESD implementation requires adaptation of this trie, rather than explicitly generating all types of label matches. Figure 1 shows rule tries used by the decoder to find matching rules during decoding, for three cases: (a) HIERO, (b) labeled system with strict matching, and (c) labeled system with ESD. Note that for (a) there are no labels except the default label "X" and the glue rule label "GOAL". In (b), labels are present both in the internal nodes and also in the leaf nodes containing the complete rules. Note that the rules "$S \rightarrow \langle NP_{\square}$ *marche lentement,* $NP_{\square}$ *walks slowly*$\rangle$" and "$S \rightarrow \langle N_{\square}$ *marche lentement,* $N_{\square}$ *walks calmly*$\rangle$" which are identical on the source-side except for their right-hand side nonterminal label (NP versus N), have distinct paths in the trie. For ESD, labels are not used as constraints and therefore need to be removed. This allows the decoder to quickly find all matching rules for a sequence of nonterminals and lexical items, without unnecessarily splitting the trie into many paths for different labelings. However, when during cube-pruning complete rules are added to the chart, the labels should still be obtainable from



(a) Rule trie for HIERO.



(b) Rule trie for a labeled system with strict label matching.



(c) Rule trie for a labeled system with soft label matching (ESD).

Figure 1: Rule tries for three different system types.

them because they are required for computing label-substitution features. This is achieved by keeping the labels outside the trie nodes but retaining them inside the rules that are stored in lists at each leaf node in the trie. This is exactly what is done in the trie for ESD (Figure 1 (c)). As an illustration, note how in (c) the rules "$S \rightarrow \langle NP_{\boxed{1}}$ *marche lentement,* $NP_{\boxed{1}}$ *walks slowly*$\rangle$" and "$S \rightarrow \langle N_{\boxed{1}}$ *marche lentement,* $N_{\boxed{1}}$ *walks slowly*$\rangle$" share the same unlabeled path in the trie as in (a), but their labels are still retained in the complete rules stored at the leaf node.

During decoding, ESD extends hypotheses with all rules matching the source input, while ignoring labels. This is done by substituting the actual labels from the hypergraph with surrogate "X" labels and using those labels to retrieve matching rules from the rule trie. There is, however, an important exception to this that requires special treatment, namely the nonterminal label occurring in glue rules. Glue rules of the form

$$GOAL \rightarrow \langle GOAL_{\boxed{1}}\, X_{\boxed{2}},\ GOAL_{\boxed{1}}\, X_{\boxed{2}} \rangle$$

contain two types of labels. The "X" symbol in the rule is the symbol that will be substituted to HIERO (non-glue) rules. The $GOAL$ label, occurring on the left-hand side of the rule and as the first nonterminal on the right-hand side, is also known as the start symbol. It serves to start the gluing extension and allows for the glue rule to be used repeatedly. This $GOAL$ label needs to be strictly matched, to prevent the left-hand side of glue rules from softly matching other nonterminals and hence substituting for HIERO rules. The strict matching of the $GOAL$ label is achieved in the grammar by retaining it as a label in the trie used by ESD (see the "GOAL" labeled internal node in Figure 1 (c), the third child of ROOT), and requiring the $GOAL$ symbols observed in the hypergraph to be strictly matched against the symbols in the trie. Furthermore, labels inside HIERO rules should not be allowed to match the $GOAL$ label but only the surrogate label X that represents the rest of the labels, when retrieving matching rules from the trie. This implementation ensures correct and efficient rule matching given either $GOAL$ labels (strict-matching) or other labels (ESD).

One other important detail enabling efficient ESD decoding is that the used labeled ESD grammars are identical in size to HIERO. Let the HIERO-*rule-signature* of a labeled rule be that rule with the labels removed. Given a rule labeling scheme, grammars used with ESD are formed by labeling every HIERO rule with a single *canonical* labeling: the most frequent labeled version across extracted rules that share the HIERO-*rule-signature* of that rule. These grammars also use the same feature set as HIERO, only adding *label-substitution features*. In contrast, because strict matching systems combine all differently labeled extracted rule versions, they use grammars that are much bigger than HIERO grammars.

4 Experiments

We evaluate our models on Chinese–English, since it facilitates the best comparison with experiments in earlier work. All data is lowercased as a last pre-processing step. The training data for our experiments is formed by combining the full sentence-aligned *MultiUN* (Eisele and Chen, 2010; Tiedemann, 2012)[2] parallel corpus with the full sentence-aligned *Hong Kong Parallel Text* parallel corpus from the Linguistic Data Consortium.[3] We used a maximum sentence length of 40 for filtering the training data. The combined dataset has 7,340,000 sentence pairs. For the development and test sets we use the *Multiple-Translation Chinese* datasets from LDC, parts 1–4,[4] which contain sentences from the News domain. We combined parts 2 and 3 to form the development set (1,813 sentence pairs) and parts 1 and 4 to form the

[2]Freely available from http://opus.lingfil.uu.se/

[3]The LDC catalog number of this dataset is LDC2004T08.

[4]LDC catalog numbers: LDC2002T01, DC2003T17, LDC2004T07 and LDC2004T07.

test set (1,912 sentence pairs). For both development and testing we use 4 references. For these experiments both the baseline and our method use a 4-gram language model with Kneser-Ney smoothing (Kneser and Ney, 1995) trained on 5,427,696 sentences of *domain-specific*[5] news data taken from the "Xinhua" subcorpus of LDC's English Gigaword corpus.[6]

4.1 Training and decoding details

Our experiments use Joshua (Ganitkevitch et al., 2012) with Viterbi best derivation. Baseline experiments use normal decoding, whereas ESD experiments relax the label-matching constraints while adding label-substitution features to facilitate learning of label-substitution preferences.

For training we use standard HIERO grammar extraction constraints (Chiang, 2007) (phrase pairs with source spans up to 10 words; abstract rules are forbidden). During decoding a maximum span of 10 words on the source side is maintained. In our experiments, for HIERO we use a standard feature set that is comparable to that of Chiang (2005). We follow Chiang (2010) in using, except for the label-substitution features, exactly the same features for ESD as for HIERO. This includes the usage of phrase-weights taken from the HIERO (label-stripped) rules as opposed to the labeled rules. For the labeled systems with strict matching (-Str), we follow Zollmann (2011) in using phrase weights for the labeled versions of the rules, but also adding smoothed versions of these features, including the HIERO (unlabeled) phrase weights. We train our systems using (batch k-best) MIRA (Cherry and Foster, 2012) as borrowed by Joshua from the Moses codebase, allowing up to 30 tuning iterations. Following standard practice, we tune on BLEU (Papineni et al., 2002), and after tuning we use the configuration with the highest scores on the development set with actual (corpus-level) BLEU evaluation. We report lowercase BLEU, METEOR (Denkowski and Lavie, 2011), BEER (Stanojević and Sima'an, 2014) and TER (Snover et al., 2006) scores for the test set. We also report average translation length as a percentage of the reference length for all systems.

To counter unreliable conclusions due to optimizer variance, we repeated all experiments three times (tuning plus testing), and compute the scores as averages over these runs; using Multeval Clark et al. (2011) version 0.5.1.[7] We also use MultEval's implementation of statistical significance testing between systems, which is based on multiple optimizer runs and approximate randomization. Differences that are statistically significant with respect to a HIERO baseline and correspond to improvement/worsening are marked with $\triangle H/\triangledown H$ at the $p \leq .05$ level and $\blacktriangle H/\blacktriangledown H$ at the $p \leq .01$ level. For average translation length, where either higher or lower may be better, we use $\square H/\blacksquare H$ to mark significant *change* with respect to the baseline at the $p \leq .05$ / $p \leq .01$ level.

We also report the Kendall reordering score (KRS), which is the reordering-only variant of the LR-score (Birch et al., 2010) (without the optional interpolation with BLEU) and which is a sentence-level score. For the computation of statistical significance of this metric we use our own implementation of the *sign test* (Dixon and Mood, 1946), described also by Koehn (2010).

Finally we report the average CPU time per translated sentence in the test set. These times are obtained using special Java system methods, and aggregated over all decoder threads and the main thread. These time statistics are robust to variations in the number of decoder threads and the amount of other jobs running on the server, factors that can easily confound statistics based on regular wall-clock time.

[5]The different domain of the training data (mainly parliament) and development/test data (news) requires usage of a domain-specific language model to obtain optimal results.

[6]The LDC catalog number of this dataset is LDC2003T05.

[7]https://github.com/jhclark/multeval

System Name	BLEU ↑	METEOR ↑	BEER ↑	TER ↓	KRS ↑	Length	CPU time
HIERO	31.63	30.56	13.15	**59.28**	58.03	97.15	3.34
HIERO-0^{th}-Str	31.90 ▲H	30.79 ▲H	13.45	60.11 ▼H	59.68 ▲H	98.65 ■H	2.87
HIERO -0^{th}	32.03▲H	30.70▲H▼S	13.42▲H	59.58▼H▲S	58.87▲H▼S	97.87■H■S	8.99
HIERO-1^{st}-Str	31.77	30.62	13.20	60.13 ▼H	59.89 ▲H	98.47 ■H	4.63
HIERO -1^{st}	32.35▲H▲S	30.98▲H▲S	13.75▲H▲S	60.26▼H	60.01▲H	99.11■H▲S	8.45
SAMT-Str	31.87 △H	30.61	13.38	59.97 ▼H	59.94 ▲H	98.46 ■H	25.59
SAMT	32.40▲H▲S	31.20▲H▲S	14.01▲H▲S	60.19▼H▽S	60.38▲H△S	99.37■H■H	8.09
BoundaryTag-Str	32.26▲H	30.94▲H	13.91▲H	60.20▼H	58.78▲H	**98.98**▲H	29.29
BoundaryTag	**32.77**▲H▲S	**31.27**▲H▲S	**14.17**▲H▲S	60.15▼H	**60.83**▲H▲S	99.72■H■S	8.60

Table 1: Results for labeled systems with strict or soft label matching. Statistical significance is given against the HIERO baseline (H) and pair-wise for every soft-matching system against its strict-matching variant (-Str). Statistical significance for the latter comparison is marked with (S). For every experiment we use boldface to accentuate the highest score across systems for all metrics, with for TER, an error metric, the lowest score instead. For length we boldface the value that is closest to 100, in absolute terms.

4.2 Is soft label matching always superior to strict matching?

In Table 1 we compare four labeled systems for decoding with strict matching and decoding with soft label matching. This extends the earlier comparison by Maillette de Buy Wenniger and Sima'an (2014, 2016), attempting to give a more general answer to the question as to whether soft label matching is always superior to strict matching. The first two systems are reordering labeled systems (Maillette de Buy Wenniger and Sima'an, 2013, 2014, 2016), and the last two systems are syntactically labeled systems, namely SAMT (Zollmann and Venugopal, 2006) and a target-side boundary-tag labeled system (Zollmann, 2011; Zollmann and Vogel, 2011).

Label Types: Our reordering labels are heuristic labels, created using hierarchical reordering information induced from word alignments. These labels come in two forms:1) 0^{th}-order reordering labels (HIERO-0^{th}) describe for each nonterminal the reordering that happens at its child nonterminals, 2) 1^{st}-order reordering labels (HIERO-1^{st}) describe the reordering of the nonterminal itself relative to an embedding parent nonterminal. SAMT is a heuristic syntactic labeling scheme, similar in spirit to combinatory categorial grammar (CCG) (Steedman, 1987, 2000). SAMT uses constituency-parse information and finds the simplest syntactic label describing a (target) span. Similar to SAMT, target-side boundary-tags are heuristic syntactic labels formed by combining the POS-tags of (target) words at the boundaries of phrase pairs. Since limited space allows only a short description of the used labels and systems, we refer the reader to the original papers for more details.

For three of the four label types tested, the soft-labeled system gives significantly better scores for BLEU, METEOR and BEER. Only the HIERO-0^{th} label type does not show significantly better results for those metrics. However, later in this section we discuss how these results too are improved by extending the search. Although it is not statistically significant, HIERO-0^{th} still shows improved BLEU for the soft-labeled version over the strict matching system. For SAMT and the target-side boundary-tag labeled system, apart from the other improvements, there are also significant improvements of KRS. The results show that soft matching is typically, although not always significantly, better than strict matching.

4.3 Challenges of soft label matching

In the previous section we saw that decoding with soft label matching typically outperforms both unlabeled systems and systems that use labels with strict matching. Nevertheless, efficient soft label matching faces two challenges: (i) increased search space, and (ii) label matching blindness.

The addition of labels increases the search space dramatically, even with soft matching.

During decoding, a rule is applied to extend an existing hypothesis. Since in practice decoding proceeds bottom up, this means any right-hand-side nonterminal label(s) of the rule are matched with the labels of the corresponding substituted to nonterminal gaps in the chart. With N labels, there are there are potentially N alternative versions per language model state in the chart entries of each gap. For a rule with 2 gaps, this means a particular rule substitution only covers one of up to N^2 possible options arising from the splitting of the language model states by labeling. Because soft matching keeps only one labeled version per HIERO-*rule-signature*, on the rule side the number of options does not increase compared to HIERO.[8]

The second challenge, which we call *label matching blindness*, is that the type of applied label substitutions (and whether or not these are matching) is only evaluated late in the search process. During initialization, cube pruning explores the combination of the best rule with the best leaf nodes for the matched-to rule gaps in the chart. However, the quality of the rule and the leaf nodes is only computed based on local (stateless) features, such as lexical probabilities and phrase weights. Features such as the language model cost cannot be computed because they cross rule boundaries. They may still be approximated given the available information, but this is generally inaccurate. In the case of *label-substitution features*, no meaningful stateless computation is possible. These features are therefore simply ignored until the *rule-nonterminal to labeled-gap* substitutions have already been decided during cube-pruning initialization.

4.4 Enriching the search

The challenges resulting from soft label matching mentioned in the last section motivate enrichment of the search during soft label matching decoding. A crude approach is to just extend the search space by increasing the value of the decoder parameter *pop-limit*, which controls the number of hypotheses that are added to the stack by cube pruning during decoding. This may improve the quality of the produced translations at the price of a higher computational cost. However, this approach is computationally expensive and inefficient, since it does not directly target an exploration of rule-to-gap substitutions with diverse labels. The initial label substitution diversity is determined by the diversity of label pairs within the set of *rule-RHS-nonterminal to chart-gap* substitutions explored during cube-pruning *initialization*. For small label sets it is feasible to enrich the set of those initially explored substitutions, thereby drastically increasing this diversity. Three ways to implement this are:

a) Exploring all alternatively labeled versions of a HIERO *source* rule type.[9]

b) Exploring all alternatively labeled gap substitutions, given the single best labeled version of a HIERO *source* rule type.

c) Combining (a) and (b), i.e. exploring all labeled rule versions and for each of those all alternatively labeled gap substitutions.

Why should this help? Assume that we explore all alternatively labeled rule versions for a HIERO rule type (a), while keeping the gap labels fixed to those of the best language model state. Then, a matching substitution will be explored if yielded by substituting the nonterminal labels for any of these alternatively labeled rule versions to those fixed gap labels. Similarly for

[8]However, with larger label sets, the canonical labeled rule form represents a larger number of rules, and will consequently be a more approximate representation of those.

[9]Note that whereas there is only one canonical labeled version per HIERO-*rule-signature*, there are potentially many labeled versions of the *source* rule type, in which the target side is ignored. Furthermore, the LHS label of the rule is ignored when collecting the best alternatively labeled versions given a HIERO rule *source* side, since it has no effect on label matching in bottom-up decoding.

System Name	Pop-limit	Explore all rule labelings	BLEU ↑	METEOR ↑	BEER ↑	TER ↓	KRS ↑	Length	CPU time
HIERO	1000	——–	31.63	30.56	13.15	59.28	58.03	97.15	3.34
HIERO 0^{th} (B_0)	1000	NO	32.03▲H	30.70▲H	13.42▲H	59.58▼H	58.87▲H	97.87■H	8.99
HIERO 0^{th}	2000	NO	32.24▲H▲B_0	30.66▲H	13.41	**59.01**▲H▲B_0	58.59△H	97.42■H■B_0	16.13
HIERO 0^{th}	4000	NO	32.30▲H▲B_0	30.85▲H▲B_0	13.71▲H▲B_0	59.56▼H	59.23▲H△B_0	98.19■H■B_0	28.27
HIERO 0^{th}	1000	YES	32.18▲H	30.77▲H△B_0	13.55▲H△B_0	59.23 ▲B_0	58.74▲H	97.69■H■B_0	9.32
HIERO 0^{th}-Sh	1000	YES	32.25▲H▲B_0	30.80▲H▲B_0	13.60▲H▲B_0	59.44▽H	59.12▲H	97.99■H	10.14
HIERO 0^{th}-Sh	2000	YES	32.47▲H▲B_0	30.87▲H▲B_0	13.69▲H▲B_0	59.51▼H	59.27▲H△B_0	98.27■H■B_0	16.59
HIERO 0^{th}-Sh	4000	YES	32.26▲H▲B_0	30.77▲H△B_0	13.55▲H△B_0	59.12△H▲B_0	58.72▲H	97.55■H■B_0	37.75
HIERO 1^{st} (B_1)	1000	NO	32.35▲H	30.98▲H	13.75▲H	60.26▼H	**60.01**▲H	99.11■H	8.45
HIERO 1^{st}	2000	NO	32.40▲H	31.00▲H	13.74▲H	60.36▼H	59.93▲H	99.15■H	15.40
HIERO 1^{st}	4000	NO	32.49▲H	30.95▲H	13.75▲H	60.29▼H	60.00▲H	99.15■H	28.84
HIERO 1^{st}	1000	YES	32.50▲H△B_1	31.03▲H△B_1	13.80▲H	60.14▼H	59.77▲H	99.05■H	9.55
HIERO 1^{st}-Sh	1000	YES	32.66▲H▲B_1	**31.05**▲H▲B_1	**13.88**▲H	60.01▼H▲B_1	59.99▲H	99.14■H	10.16
HIERO 1^{st}-Sh	2000	YES	32.62▲H▲B_1	30.90▲H▼B_1	13.70▲H	59.72▼H▲B_1	59.41▲H▼B_1	98.60■H■B_1	21.79
HIERO 1^{st}-Sh	4000	YES	**32.70**▲H▲B_1	30.99▲H	13.83▲H	60.22▼H	59.91▲H	**99.20**■H	40.81

Table 2: The effects on the results of: 1) the pop-limit, 2) exploring all alternative rule labelings, with or without shuffling. Statistical significance is given against the HIERO baseline (H) and for each label against the basic search version: HIERO 0^{th} without additional search (B_0) or HIERO 1^{st} without additional search (B_1).

System Name	Pop-limit	Explore all rule labelings	BLEU ↑	METEOR ↑	BEER ↑	TER ↓	KRS ↑	Length
HIERO	1000	——–	31.70	30.72	13.40	61.21	58.28	99.49
HIERO 0^{th} (B_0)	1000	NO	31.85▲H	30.82▲H	13.78▲H	61.46▼H	59.08▲H	**100.00**■H
HIERO 0^{th}	2000	NO	32.08▲H	30.76	13.77▲H	**60.86**▲H	58.92▲H	99.61
HIERO 0^{th}	4000	NO	32.03▲H	30.97▲H	13.94▲H	61.47▼H	59.42▲H	100.36■H
HIERO 0^{th}	1000	YES	32.05▲H	30.86▲H	13.72△H	61.09	59.05▲H	99.93■H
HIERO 0^{th}-Sh	1000	YES	32.01▲H	30.92▲H	13.82▲H	61.33	59.53▲H	100.19■H
HIERO 0^{th}-Sh	2000	YES	32.12▲H	30.97▲H	13.93▲H	61.42▽H	59.53▲H	100.23■H
HIERO 0^{th}-Sh	4000	YES	32.07▲H	30.90▲H	13.83▲H	60.99△H	58.95▲H	99.85■H
HIERO 1^{st} (B_1)	1000	NO	32.13▲H	31.22▲H	14.21▲H	62.31▼H	**60.61**▲H	101.29■H
HIERO 1^{st}	2000	NO	32.43▲H	31.24▲H	14.24▲H	62.20▼H	60.43▲H	101.33■H
HIERO 1^{st}	4000	NO	32.46▲H	31.26▲H	14.30▲H	62.02▼H	**60.61**▲H	101.35■H
HIERO 1^{st}	1000	YES	32.33▲H	31.22▲H	14.32▲H	62.12▼H	60.32▲H	101.15■H
HIERO 1^{st}-Sh	1000	YES	32.38▲H	31.27▲H	**14.37**▲H	62.05▼H	60.29▲H	101.19■H
HIERO 1^{st}-Sh	2000	YES	32.48▲H	31.09▲H	14.10▲H	61.65▼H	59.85▲H	100.73■H
HIERO 1^{st}-Sh	4000	YES	**32.55**▲H	**31.30**▲H	**14.37**▲H	62.20▼H	60.55▲H	101.49■H

Table 3: Analysis: results on the **development** set. We demonstrate the effect on the results of: 1) the pop-limit, 2) exploring all alternative rule labelings, with or without shuffling. Statistical significance is given against the HIERO baseline (H).

Evaluation order without shuffling:

1. S -> [NP,1] marche lentement, [NP,1] walks slowly (A)
2. S -> [NP,1] marche lentement, [NP,1] strolls (B)
3. S -> [N,1] marche lentement, [N,1] walks calmly (C)
4. **S -> Elle marche [ADV,1], She walks [ADV,1] (D)**

Final order after scoring:

1. S -> [NP,1] marche lentement, [NP,1] walks slowly |||
   Elle marche lentement / She walks slowly ||| -4.0 (A)
2. S -> [NP,1] marche lentement, [NP,1] strolls |||
   Elle marche lentement / She strolls ||| -5.0 (B)
3. S -> [N,1] marche lentement, [N,1] walks calmly |||
   Elle marche lentement / She walks calmly ||| -5.0 (C)
4. **S -> Elle marche [ADV,1], She walks [ADV,1]**
   Elle marche lentement / She walks leisurely ||| -5.0 (D)

(a) Evaluation order and final order without shuffling.

Evaluation order with shuffling (i.e. random):

1. **S -> Elle marche [ADV,1], She walks [ADV,1]** (D)
2. S -> [NP,1] marche lentement, [NP,1] walks slowly (A)
3. S -> [N,1] marche lentement, [N,1] walks calmly (C)
4. S -> [NP,1] marche lentement, [NP,1] strolls (B)

Final order after scoring:

1. S -> [NP,1] marche lentement, [NP,1] walks slowly |||
   Elle marche lentement / She walks slowly ||| -4.0 (A)
2. **S -> Elle marche [ADV,1], She walks [ADV,1]**
   Elle marche lentement / She walks leisurely ||| -5.0 (D)
3. S -> [N,1] marche lentement, [N,1] walks calmly |||
   Elle marche lentement / She walks calmly ||| -5.0 (C)
4. S -> [NP,1] marche lentement, [NP,1] strolls |||
   Elle marche lentement / She strolls ||| -5.0 (B)

(b) Evaluation order and final order with shuffling.

Figure 2: Example of the effect of shuffling on the decoding, translating the source phrase "Elle marche lentement".

(b), given a single fixed labeled rule version, a matching substitution is explored if achievable in combination with the available gap labelings.

With respect to computational complexity, (a) and (b) potentially increase the number of explored combinations by a factor of N^2 with N being the size of the label set, whereas (c) even increases it by N^4. These increases are with respect to the number of applicable HIERO rule types, since the increased exploration only concerns the cube pruning initialization and not the whole cube pruning process. Consequently, if N is small, the empirical increase in computational cost is limited. With larger N however, the increase in computational cost of the initialization (quadratic for (a) and (b) and N^4 for (c)), starts to dominate the total cost, so that none of these approaches scale to large[10] label sets, with (c) scaling up worst.

Ie what follows, we only explore (a), which is very similar to (b). We will refer to this setting as diverse rule labeling exploration (DRLE) in the rest of the paper. We do not explore (c) here, because of its very restricted scalability.

4.5 Shuffling

In cube-pruning initialization, applicable rules are substituted to particular gaps, and every complete rule substitution leads to a total score that includes the label-substitution features and language-model cost amongst other things. These complete rule substitutions are then added as initial options to the cube-pruning queue.

In our initial implementation of DRLE (see Section 4.4), we consecutively evaluated all alternative labeled versions for a specific HIERO source-rule type before moving on to the next type. Independent of this evaluation order, rules are placed on the cube-pruning queue in the order of their evaluation scores. Nevertheless, we discovered that it helps if we shuffle the order of the rules before we evaluate them. This might seem odd, since this shuffling (*-Sh*) can only affect the order of rules yielding the same score. However, that is exactly how we think shuffling helps; without shuffling, all labeled versions of the same rule source-side with the same score are lumped together in the cube pruning queue. Shuffling mixes them with other rules that tie for the same score. This increases diversity when neither the labels nor the translation for a rule source-side are discriminative, as is common for certain rules at the start of tuning when

[10]The approaches were still feasible for at least $N = 25$ labels, the highest number we have tested. We acknowledge however, that once N increases significantly, problems will be encountered due to computational complexity.

label-substitution feature weights are initialized to zero.

To better understand shuffling and how it effects the order of hypotheses in the cube-pruning queue, it is helpful to look at an example. In Figure 2 we show the effect of shuffling on the translation of the French source phase "Elle marche lentmentent". Here, we assume that the first source word ("Elle") and last source word ("lentement") have already been translated before. Hence, a full translation can be formed by combining these previous translations with HIERO rules that additionally translate the first two words ("Elle marche") or last two words ("marche lentement") respectively, substituting the earlier translated last/first word as a gap. In Figure 2a we show the evaluation order and final order of hypotheses without shuffling. As mentioned before, without shuffling all rules that share the same source-side (ignoring labels) – in this case rules A,B and C – are evaluated consecutively. Then in the next step the scored rules are sorted by their score, which in this example does not further change the order. In Figure 2b in contrast, the rules are shuffled, randomly permuting their order before evaluation. In this case the random order of rule evaluation is D,A,C,B. Then after scoring, rule A again comes to the top, because it has the highest score. The relative order of D,C and B however remains changed because rules B, C and D tie for the same score, so their relative order before scoring determines their relative final order. Notice in particular how rule D (in boldface), which has a different source side from rules A, B and C, now comes directly after rule A in the final order.

Note that shuffling only randomizes the relative order in which rules tying for the same score are added to the cube-pruning queue, eliminating implementation-specific bias for the order of such rules. This avoids different labeled versions of the same rule with the same score all clinging together in the queue as an undesirable side-effect of the specifics of the DRLE implementation. Because shuffling randomizes the order of rules with the same score in the final queue, it also removes the opportunity for the tuner to lazily exploit partially deterministic order in development set hypotheses which is of no use for translation of the test set. Possibly, this by itself also has a positive effect in making tuning more robust and reducing the chance of overfitting. We leave it for future work to further investigate this. Crucially, shuffling does not specifically add additional search errors.

4.6 Effects of search extension strategies

Table 2 shows the effects of the different strategies described in the previous sections to expand the search space. The table first repeats the results for the HIERO baseline and then lists results for the HIERO 1^{st} and HIERO 0^{th} reordering labels. For each we then use either the standard setting for rule exploration during cube-pruning initialization, or DRLE. We do this in combination with shuffling, and also vary the pop-limit. This provides insight into the effect of these factors when applied independently or combined.

In addition to the HIERO baseline, our second baseline (B_0/B_1) for each of the two separate reordering labels is a basic ESD system without changes to the default pop-limit (1000) and without DRLE. For each reordering label we then test the significance of improvements against both HIERO and B_0/B_1. We see that both increasing the pop-limit and DRLE improves results for both label types. However, when only the pop-limit is increased without DRLE, HIERO 0^{th} improves significantly over the basic (B_0) system, while HIERO 1^{st} gives no significant improvements over the basic (B_1) system. In addition, for HIERO 1^{st}, comparing the effect of just using DRLE to just changing the pop-limit, the former outperforms the latter for all metrics except KRS over all tested values of the pop-limit. Adding shuffling makes this trend is even sharper. For both label types, shuffling has a positive effect on the results. Note too that DRLE has the highest impact with larger label sets; probably the resulting larger search space increases the chance of missing certain desirable label substitutions in the normal cube pruning initialization.

System Name	BLEU ↑	METEOR ↑	BEER ↑	TER ↓	KRS ↑	Length	CPU time
HIERO	31.63	30.56	13.15	**59.28**	58.03	97.15	3.34
HIERO 0^{th}	32.50▲H	30.88▲H	13.68▲H	59.66▼H	59.30▲H	98.40■H	29.50

Table 4: Analysis experiment: tuning with pop-limit 2000 and test-set decoding with pop-limit 4000.

Summarizing the results over both label types, we can conclude that DRLE is typically better than just crudely increasing the pop-limit. Additionally, for small label sets it comes at a lower computational cost.

4.7 Analysis: negative interaction DRLE with higher pop-limit explained.

Both DRLE and an increased pop limit by themselves have a positive effect on the translation quality, but a surprising result is the sometimes relatively negative effect of using DRLE and also maximally increasing the pop-limit. With HIERO 0^{th} the results improve when increasing the pop-limit to 2000, but then drop when further increasing it to 4000. For HIERO 1^{st}, with DRLE only BLEU benefits slightly from a pop-limit higher than 1000, whereas performance decreases for the other metrics. Such negative interactions between DRLE and the increased pop-limit could be caused by overfitting.[11] To see if overfitting indeed occurs, we looked at the evaluation scores for the development set, see Table 3. It can be seen that for HIERO 0^{th} with DRLE, the BLEU scores decrease when the pop-limit increases from 2000 and 4000, but in particular the decrease in the development set BLEU score is less than the decrease in the test set BLEU score, see Table 2. Furthermore, when looking at HIERO 1^{st} with DRLE on the development set, it can be seen that the BLEU score monotonically increases for increasing pop-limit size. However, other metrics show a dip for a pop-limit of 2000, which was also seen for the test set for all metrics except TER. To summarize, for certain increases in the pop-limit in combination with DRLE, we made two observations that indicate overfitting:

- Loss of performance on the test set, for most metrics including BLEU, the tuning metric.
- Mostly retained or even increased performance for BLEU on the development set, combined with performance loss for most other metrics.

Could it still be that a higher pop-limit is by itself harmful, independent of its role in the assumed overfitting? We hypothesize that it is only harmful in as far as it facilitates overfitting in combination with DRLE during the tuning process. To test this hypothesis we ran another analysis experiment, whereby we use the final feature weights obtained from tuning with DRLE with a pop-limit of 2000 and only increase the pop-limit to 4000 during the decoding of the test set. The results are shown in Table 4. As can be seen, in this setting the results are highly similar to the results obtained with DRLE and a pop-limit of 2000 used for both tuning and testing. This confirms our hypothesis that a higher pop-limit (more search) is not generally harmful, but can be harmful in the tuning stage because it facilitates more overfitting.

4.8 Combining Labels

In this section we look at the effect of combining multiple labels. The first successful combination we explore is 0^{th}-order and 1^{st}-order reordering labels. Since both labels individually give good results, and encode somewhat different information about word order, their combination could work even better. The other two combinations we test are 1^{th}-order reordering labels combined with SAMT or target-side boundary-tags. These combinations are

[11] DRLE groups the translations of the source side by their labels and uses the best translation for each distinct labeling, as opposed to only a single best translation. This may cause also more suboptimal source rule-side translations to be added to the initial cube-pruning queue; and with a higher pop-limit there is a higher chance of those being retained and causing problems such as overfitting.

System Name	BLEU ↑	METEOR ↑	BEER ↑	TER ↓	KRS ↑	Length	CPU time
HIERO	31.63	30.56	13.15	**59.28**	58.03	97.15	3.34
HIERO 0^{th} + HIERO 1^{st}	32.30▲H▲H_0	30.95▲H▲H_0	13.75▲H▲H_0	60.16▼H▼H_0	60.13▲H▲H_0	99.07▲H■H_0	7.18
SAMT+ HIERO 1^{st}	32.57▲H△H_1	31.07▲H▼S▲H_1	13.84▲H▽S	59.94▼H▲H_1	60.18▲H	99.13■H■S	11.63
Bnd.Tags+ HIERO 1^{st}	**32.65**▲H▲H_1	**31.36**▲H▲B▲H_1	**14.16**▲H▲H_1	60.21▼H	**61.46**▲H▲B▲H_1	**99.86**■H■H_1	10.32

Table 5: Double-labeled systems with soft matching. Result are for exploring only the best rule labeling and label substitution during cube pruning initialization. Statistical significance is given against the HIERO baseline (H) and for every double-labeled system against the single-labeled systems from which the double label is composed: HIERO 0^{th} (H_0), HIERO 1^{st} (H_1), SAMT (S) and target-side boundary-tags (B).

intuitively promising since reordering labels and syntactic labels are expected to give at least partially different information that may be expected to be complementary.

When combined labels are directly applied to form label-substitution features, this yields a quadratic increase in the number of these features, causing extreme sparsity and hence overfitting problems. We thus take another approach; during feature generation, we split every combined label into its two constituent parts and compute individual label-substitution features for each. Table 5 shows the results of the label-combination experiments. Most of the double-labeled systems come to the level of the best of the two constituent labels, but do not improve beyond it. However, the system that combines target-side boundary-tags and 1^{th}-order reordering labels significantly improves over both these labels individually for both METEOR and KRS. These metrics are particularly concerned with assessing the quality of the word order, which receives less or no attention in the other metrics. Since reordering labels are particularly expected to improve word order, it is positive that they help to further improve it for the best-performing single label in our experiments.

5 Conclusion

In this work, we examined key aspects of effective and efficient ESD. We first gave a detailed description of how this method can be efficiently implemented, and then examined three empirical questions. First, based on experiments for four different label types, we demonstrated that ESD is empirically at least equal but typically superior to strict matching. Next, we demonstrated that ESD can benefit from richer search. Our experiments show that it is more effective to specifically target the search effort towards the exploration of more diverse label substitutions instead of crudely increasing search in general by using a higher pop-limit. Finally, we explored the effect of double labels, and showed that while these are not successful in general, the specific combination of target-side boundary-tags and reordering labels does significantly improve word order as measured by METEOR and KRS, without significantly changing the other metrics.

Acknowledgements

This research is supported by the ADAPT Centre for Digital Content Technology, funded under the SFI Research Centres Programme (Grant 13/RC/2106). This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Marie Skłodowska-Curie grant agreement No 713567. The investigations were supported by The Netherlands Organization for Scientific Research (NWO) under grant nr. 612.066.929 and VICI grant nr. 277-89-002 and Stichting voor de Technische Wetenschappen (STW) grant nr. 12271. We would like to thank the anonymous reviewers for their helpful comments.

References

Aho, A. V. and Ullman, J. D. (1969). Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3(1):37–56.

Almaghout, H., Jiang, J., and Way, A. (2011). CCG contextual labels in hierarchical phrase-based smt. In *Proceedings of the 15th Annual Conference of the European Association for Machine Translation (EAMT-2011)*, pages 281–288, Leuven, Belgium.

Birch, A., Osborne, M., and Blunsom, P. (2010). Metrics for MT evaluation: Evaluating reordering. *Machine Translation*, 24(1):15–26.

Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montreal, Canada.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 263–270, Ann Arbor, Michigan.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Chiang, D. (2010). Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1443–1452, Uppsala, Sweden.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, pages 176–181, Portland, Oregon.

De La Briandais, R. (1959). File searching using variable length keys. In *Papers Presented at the the March 3-5, 1959, Western Joint Computer Conference*, IRE-AIEE-ACM '59 (Western), pages 295–298, San Francisco, California.

Denkowski, M. and Lavie, A. (2011). Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland.

Dixon, W. J. and Mood, A. M. (1946). The statistical sign test. *Journal of the American Statistical Association*, 41(236):557–566.

Eisele, A. and Chen, Y. (2010). Multiun: A multilingual corpus from united nation documents. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 2868–2872, La Valletta, Malta.

Ganitkevitch, J., Cao, Y., Weese, J., Post, M., and Callison-Burch, C. (2012). Joshua 4.0: Packing, pro, and paraphrases. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 283–291, Montréal, Canada.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 181–184, Detroit, Michigan, USA.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Edmonton, Canada.

Li, J., Tu, Z., Zhou, G., and van Genabith, J. (2012). Using syntactic head information in hierarchical phrase-based translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 232–242, Montréal, Canada.

Maillette de Buy Wenniger, G. and Sima'an, K. (2013). Hierarchical alignment decomposition labels for hiero grammar rules. In *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 19–28, Atlanta, Georgia, USA.

Maillette de Buy Wenniger, G. and Sima'an, K. (2014). Bilingual markov reordering labels for hierarchical SMT. In *Proceedings of the Eight Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 11–21, Doha, Quatar.

Maillette de Buy Wenniger, G. and Sima'an, K. (2016). Labeling hiero grammars without linguistic resources. *Machine Translation*, pages 1–41.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

Stanojević, M. and Sima'an, K. (2014). BEER: BEtter evaluation as ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419, Baltimore, Maryland, USA.

Steedman, M. (1987). Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory*, 5:403–439.

Steedman, M. (2000). *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2868–2872, Istanbul, Turkey.

Venugopal, A., Zollmann, A., Smith, N. A., and Vogel, S. (2009). Preference grammars: softening syntactic constraints to improve statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 236–244, Boulder, Colorado.

Zollmann, A. (2011). *Learning Multiple-Nonterminal Synchronous Grammars for Statistical Machine Translation*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA.

Zollmann, A. and Venugopal, A. (2006). Syntax augmented machine translation via chart parsing. In *NAACL 2006 - Workshop on statistical machine translation*, pages 138–141, New York City, New York, USA.

Zollmann, A. and Vogel, S. (2011). A word-class approach to labeling pscfg rules for machine translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1–11, Portland, Oregon, USA.

Development of a classifiers/quantifiers dictionary towards French-Japanese MT

Mutsuko Tomokiyo Mutsuko.Tomokiyo@imag.fr
Mathieu Mangeot Mathieu.Mangeot@imag.fr
Christian Boitet Christian.Boitet@imag.fr
UGA, GETALP-LIG, bâtiment IMAG, 700 avenue Centrale,
Domaine Universitaire de Saint-Martin-d'Hères, CS 40700
38058 Grenoble CEDEX 9, France

Abstract

Although classifiers/quantifiers (CQs) expressions appear frequently in everyday communications or written documents, they are described neither in classical bilingual paper dictionaries, nor in machine-readable dictionaries. The paper describes a CQs dictionary, edited from the corpus we have annotated, and its usage in the framework of French-Japanese machine translation (MT).

CQs treatment in MT often causes problems of lexical ambiguity, polylexical phrase recognition difficulties in analysis and doubtful output in transfer-generation, in particular for distant languages pairs like French and Japanese.

Our basic treatment of CQs is to annotate the corpus by UNL-UWs (Universal Networking Language - Universal words)[1], and then to produce a bilingual or multilingual dictionary of CQs, based on synonymy through identity of UWs.

Keywords: classifiers, quantifiers, corpus annotation, UNL, UWs dictionary, phraseology study, Tori Bank, French-Japanese MT

Introduction

We call CQs (classifiers/quantifiers) words or phrases which are used in some languages to indicate the class of a noun or a nominal phrase, depending upon the type of its referent or upon speaker's observation of the referent, when they appear in quantitative expressions. They denote:

(a) CQs expressing quantity of the referent by counting.

Eg. pièce (piece) (in French), 枚(mai, sheet), 点 (ten, piece) (in Japanese), cm, gram

(b) CQs representing quantity concept, based on speaker's observation or general metonymy.

Eg. un brin de (a little), bribes de (scraps of), ひとつまみの (hito-tsumami no, a pinch of), 山盛りの (yama-mori no, a pile of).

There are two cases for a CQ: (1) it can belong to only the (a) type or the (b) type, and (2) it can belong at the same time to both the (a) and (b) types. That is because, on the

[1]The UNL (Universal Networking Language) project was founded at the Institute of Advanced Studies (IAS) of the United Nations University in Tokyo in April 1996 under the aegis of UNU (United Nations University, Tokyo) and with financial support from ASCII corporation (a Japanese publishing company, 1977-2002) and UNL-IAS. http://www.undl.org/unlsys/unl/unl2005/attribute.htm

one hand, there are some CQs that play only the role of classifier or quantifier, and, on the other hand, there are CQs that play both of these roles.

Eg. un brin de paille (a wisp of straw), un brin de folie (a touch of madness)[2].

When we started to deal with CQs expressions in the framework of French-Japanese MT, we met mainly the following difficulties, which were inherent in QCs:

1. Resolution of lexical ambiguity of polysemic nouns
   Eg. pièce (piece) : (Japanese translation as CQs) 枚(mai, sheet or ϕ[3]), 点 (ten, ϕ), 頭(tou, ϕ), 樽(taru, cask), etc.

2. Producing adequate CQs in Japanese when they are absent in French
   Eg. deux livres (two books) : (Japanese translation) 二冊の本 (ni-satsu no hon)

   ni = two, satsu = ϕ, no = postposition, hon = book, where 冊 (satsu) is one of the Japanese CQs for books, notebooks, albums, etc.

3. Normalization for floating quantifier phenomenon in Japanese

4. Recognition of QC polylexical expressions over the course of corpus development
   Eg. une pincée de sel (a pinch of salt): (Japanese translation) ひとつまみの塩 (hito-tsumami no shio)

   hito = 1, tsumami = pinch, no = of , shio = salt

To handle these linguistic behaviours of CQs in a comprehensive manner, we have adopted the UNL-UWs format for our corpus annotations and dictionary descriptions. Another motivation is the desire to be able to extend this work to many other languages, in the framework of MT based on the passage through the UNL semantic pivot.

In this paper, we first examine the behaviour of CQs and the related problematic issues more concretely, from the point of view of French↔Japanese MT, and then propose a resolution of the above-mentioned problems by extending the UNL-UWs dictionary.

1 Lexical ambiguity for classifiers/quantifiers

According to our studies on ambiguities for MT, 14% of analysis errors are due to polysemous words[4] [Boitet and Tomokiyo (1995), Boitet and Tomokiyo (1996), Tomokiyo and Axtmeyer (1996)]. Also, Wisniewski et al. (2013) say the most frequent necessary post-editing operation in their French corpus translation into English is to correct articles like "les", "le", "du", etc., and the next one concerns lexical transfer errors of polysemous words.

We have also confirmed that, when polysemous words are used in their abstract or figurative meaning in CQs expressions, translation results produced by current MT systems are not at all good, because words contained in CQ phrases are often at the same time polysemous and are used in their figurative meaning.

The following example shows « pincée (pinch, つまみ, tsumami) » appearing in a quantifier phrase « une pincée de », and used in its figurative meaning. When one looks at the translation outputs produced by free as well as commercial MT systems, it appears that there is a lack of phraseology studies and polysemy disambiguation method for the word « pincée »[5].

[2] "brin" means (1) a small stalk, and (2) "a bit, a little" in "un brin de"

[3] The symbol ϕ means the absence of corresponding translation in French.

[4] We have carried on a research on ambiguity analysis from the lexical, semantic and contextual points of view since 1996. Ambiguities have been defined, categorized, and formalized as objects in an ambiguity database, and we have used this theoretical background to label ambiguities in Japanese-English interpreted dialogues, collected for the development of a speech translation system at ATR in Japan (1996).

[5] The word "pincée" is used as CQs in form of "une pincée de"+noun without particle" for pulverized substances.

Table 1: problem of CQ words ambiguity in French-Japanese MT

French word	Examples[7]	English translation	Japanese translation
pièce	une pièce de toile	a piece of cloth	一枚 (ichi-mai) の布
	une pièce de mobilier	a piece of furniture	一点 (it-ten) の家具
	dix pièces de bétail[8]	ten pieces of cattle	１０種 (jyut-tou) の家畜
	plusieurs pièces de bois	several pieces of wood	数枚 (suu-mai) の板
	Une pièce de vin est un tonneau de vin contenant environ 220 litres.	A cask of wine is a barrel of wine containing about 220 liters.	一樽 (hito-taru) のワインとは約２２０リットルを含むワイン樽である。
	J'ai reçu une demi-pièce de ce vin.	I received half a cask of this wine.	わたしは半樽(han-taru) のワインを受け取った。
	Dans une pièce de théâtre, il n'y a pas de narrateur pour raconter les faits.	In a play, there is no narrator to tell the facts.	ある作品 (aru-sakuhin) では事実を語るナレータが いない。
	une pièce de viande	a piece of meat	一切れの肉 (hito-kire)
	une pièce de blé	a wheat field	一枚 (ichi-mai) の麦畑 (no mugi-batake)

Eg. Ajoutez une pincée de sel. (ひとつまみの塩を加えなさい (hitotsumami-no shio-wo kuwaenasai), Add(加えなさい) a pinch of (ひとつまみの) salt (塩).) → (translation outputs) 塩のつねり (tsuneri) を加えなさい / 塩のピンチ (pinchi) を加えなさい / 塩のピンチ (pinchi) を追加します (shio no tsuneri wo kuwaenasai / shio no pinchi wo kuwaenasai / shio no pinchi wo tsuikashimasu)[6].

Even measure words like cm, km, kg, etc. have acronym ambiguity [Mari (2011)].
Eg. cm ← centimètre, congrégation de la mission, coût marginal, etc.

To disambiguate a polysemic CQ, we describe each of its meanings, with the associated conditions of occurrence, as a UW (contained in our Universal Words dictionary).

In our fr-UW dictionary, the description for the ambiguous word "pièce" is as follows:
pièce → cask(icl>wine)
pièce → piece(icl>cloth)
pièce → piece(icl>furniture)
pièce → piece(icl>meat)
pièce → room(icl>place)

[6]The translations on following MT systems don't make sense.
http://www.reverso.net/translationresults.aspx?langF̄R&directionf̄rancais-japonais.
http://www.worldlingo.com/fr/products_services/worldlingo_translator.html. https://translate.google.com/#fr/en/a

[7]The sources of the examples are the French-Japanese dictionary "Royal", the information on "pièce" in the Wiktionary "Vinothèque" article, see https://fr.wiktionary.org/wiki/pièce_de_vin, and http://www.etudes-litteraires.com/etudier-piece-de-theatre.php

[8]Each animal, like ox, cow, etc., that belongs to cattle. One says rather "head of cattle" today.

[9]The actant means here an expression that helps complete the meaning of a predicate.

[10]The semantic relation labels are created from UNL ontology, which store all relational information in a lattice structure, where UWs are interconnected through relations including hierarchical relations (10 levels) such as "icl" (a-kind-of) and "iof" (an-instance-of), and mean headword's sub-meaning and equivalent quantity, respectively. http://www.undl.org/unlexp/

Table 2: UWs and UWs dictionary

A UW is a character string of the form "headword(constraint_list)" which represents a concept associated to the headword. For example, "look(agt>thing, equ>search, icl>examine(icl>do, obj>thing))" is a possible UW for the meaning of the verb "look" corresponding to "examine". Other UWs will be used for various meanings of "look" as a noun: appearance (Paul's look(s)), or action (after a quick look,...).
The semantic representation of an utterance in UNL is a hypergraph, where each node bears a UW, possibly augmented by semantic attributes, and arcs bear semantic relations from a small list of about 40, like "agt", "obj", "aoj", "ben".
In fact, there are three types of UW: *r*estricted UWs, which are formed as said above (headword plus constraint list), *e*xtra UWs, which are a special type of restricted UWs, and *b*asic UWs, which are bare headwords, with no constraint list.
The syntax for dictionary description is:

```
<UW> ::= <Headword>['('<Constraint_List>')']
```

The constraint list restricts the interpretation of a UW to a specific concept included within those covered by the Basic UW [Uchida et al. (2006)], or to a subset of them. Eg.

look(agt>thing, equ>search, icl>examine(icl>do, obj>thing))
relever (to season): season(agt>person, obj>dish, icl>action)
樽 (taru, to cask): cask(icl>wine, equ>220 litres)

The semantic relation "agt" denotes that the first actant[9] of "look" is a "thing", "look" belongs to equivalent semantic level in UNL ontology map[10] with "search", and includes the meaning of "examine", "examine" is an action verb and its grammatical object is a noun meaning things.
The UNL-lang dictionaries contained, at the moment of writing, 1269421 headwords for Japanese, 520305 headwords for French and 1458686 headwords for English. The semantic attributes consist of 58 labels and semantic relation labels [Uchida et al. (2006)].
For French-Japanese translation, French words are converted into UWs by using a UNL-French dictionary, and a UNL-Japanese dictionary is used for generating Japanese translations.

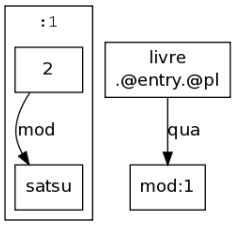
2 Handling dummy classifiers

A frequent but difficult case appears when a CQ does not appear explicitly in one language of a source-target language pair[11], nevertheless they are mandatory in type (a) CQ usage, like 冊 (satsu) for counting books, notebooks, albums, etc., 匹 (hiki) for counting small animals, 台 (dai) for counting cars, bicycles, pianos, computers, etc. Eg.
2 livres (two books) → 二冊の本 (ni-satsu no hon)
ni = 2, satsu = ϕ, no =ϕ, hon = books
un chat (a cat) → 一匹の猫 (i-ppiki no neko)
i = 1, ppiki = ϕ, no = ϕ, neko = cat

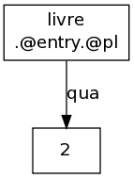
There is no lexeme in French corresponding to 冊 (satsu), but if 冊 (satsu) is omitted in the translation into Japanese, the sentence doesn't make sense. In order to represent such Japanese sentences in UNL, which is based on English, when these CQs don't exist in English, we create new UWs beginning by "CQ-<romanized Japanese CQ>", followed by a list of some English referent nouns. For example: CQ-satsu-books-notebooks-albums, "CQ-dai-cars-bicycles-computers-pianos"[12].

Absent CQs in French are marked by the attribute "@eld" (elided), which we have added to the original attribute list.

Eg. Description for 冊 (satsu) in Japanese-UW dictionary:
冊 (satsu) (icl>CQ-books, notebooks, albums)
Accordingly, the graphs for 二冊の本 (two books) is as follows:
qua(book(icl>thing).@pl, :01)
mod:01(CQ-satsu-books-notebooks-albums(icl>CQ).@entry.@eld, 2)



(a) Tentative japanized UNL-graph for "二冊の本 (two books)"



(b) Tentative frenchized UNL-graph for "deux livres (two books)"

[11]This happens not only between Japanese and western languages, but also between French and English: eg. une pièce de blé → a wheat field, une pièce de théâtre → a play

[12]At present, new CQs are made by indicating only some modifiable nouns, but this should be completed by labels coming from Mel'chuk's labels in the "Dictionnaire explicatif et combinatoire du français contemporain (DEC)" (1999, Montréal, UdM Press). In the DEC, a word is analyzed from 5 points of view: general morphosyntax, semantics, syntactic combinatorics, lexical co-occurence, phraseology. The analysis of the lexical co-occurences is made by using 60 labels corresponding to as many lexico-semantic functions (FLs) such as Magn, Anti-Magn, Mult, Sing, etc. Magn(X) is "very X", Mult(X) is "a regular quantity of X" and Sing(X) is "a regular quantum of X".

Values of FLs are subsets of lexemes, ordered by degree of intensity of the relation. For example, Magn(fever) = {high; strong; horse}, Mult(fish) = {shoal, school}, and Sing(wine) = {glass, bottle, cask, liter...}.

When possible, we will use these labels instead of the above labels such as "CQ-concrete nouns". Note that it is not possible in cases where two or more Japanese counters corresponding to different measures can apply to the same nominal concept, but don't exist in English: to use only the FL label would lead to a loss of information and to the impossibility of exact translation. Examples:
CQ-tou = [qua(mod(icl>animal, Magn), number]
CQ-piki = [qua(mod(icl>animal, Anti-Magn), number]

Table 3: Positions of numerical phrases in Japanese

Morphology	Japanese sentence and English translation	words order and word-to-word correspondance to English translation
Numerical word and CQs	本を二冊買いました (I bought two books.)	hon = book, wo = postposition(ϕ), ni = 2, satsu= ϕ, kaimashita = bought
Numerical word +CQs+の(no, of)+Noun	二冊の本を買いました(I bought two books.)	ni=2, satsu=ϕ, no = postposition(ϕ), hon = books, wo = postposition(ϕ) kaimashita = bought
Noun+Numerical word+CQs	本二冊買いました(I bought two books.)	hon = books, ni = 2, satsu = ϕ, kaimashita = bought
Numerical word+CQs	本を買いました，二冊 (I bought books, two.)	hon = books, wo = postposition(ϕ), kaimashita = bought, ,=comma, ni= 2, satsu = ϕ

3 Association of numerical phrase with its host phrase

There are two different aspects concerning the floating quantifier behaviour in Japanese [Miyagawa (1989)].

Firstly, the problem we have encountered in the process of Japanese-French MT, lies in the fact that the Japanese quantifiers can be freely positioned among phrase units in a sentence.

The "Numerical word + CQ + の (no, of) + Noun" type can be split into the CQ phrase and the «Noun» part, in which case a CQ phrase behaves like an adverb before the predicative verb in a sentence. Hence, three types of expressions are possible for the same meaning [Miyagawa (1989)].

Standardization of a floating CQ position consists in determining the CQ phrase and its host phrase, when they are separated in a sentence. In fact, the floating quantifier phenomenon exists also in French, although its linguistic behaviour is different[13] from the Japanese case. Hence, we need modifiable nouns information for each quantifier in order to find out their host noun phrase.

Secondly, there is a risk of generating meaningless expressions as a Japanese translation outputs in some cases, when the association condition between a floating CQ and its host phrase is not given. For instance, "3kg の子豚がいました" (3kg-no kobuta-ga imashita) (There was a 3kg piglet.) is acceptable as a Japanese sentence, but "子豚が3kgいました" (kobuta-ga 3kg imashita)*[14] doesn't make sense, because «子豚 (kobuta, piglet)» means only an alive piglet and co-occurs with "いました" (there was), but "3kg" cannot[15]. Hence, to avoid a machine translation output such as "子豚が3kgいました" (observed), supplementary information on "子豚" on the verb "いる" (iru, there is, or exists) and on how to use that information is necessary. For that reason, we also use a UNL-jp dictionary, which enables us to describe semantic cooccurence information between words (here, japanese lemmas).

In order to find the host phrase of a floating CQ, that is, to get the same translation results for the sentences which are morphologically different but have the same meaning,

[13]Floating CQs in French are "tous", "toutes", etc., number and gender agreement is obligatory between two phrases [Miyagawa (1989), Bobaljik (2001)], whereas there are neither number nor gender for common nouns in Japanese.

[14]子豚が3kgいました*, For the piglet, there were 3 kg*.

[15]There are two verbs expressing "existence" or "presence" in Japanese: "いる (iru)" for human being and animals and "ある" (aru) for things

we add some information to “aoj”, mentioned above in the square.
Descriptions for いる (iru) and ある (aru) are as follows.
いる (iru) : there-be(obj>animal)
いる (iru) : there-be(obj>person)
ある (aru) : there-be(obj>thing)

4 Recognition of quantifiers/classifiers and phraseology

The Type (a) CQs above-mentioned come from Phrase Book II, Tori Bank[16] (see Annex 1), while referring to existing weights and measures dictionaries[17]. Phrases book II includes basic CQs which were manually or semi-automatically collected from journals, novels, numerous articles on the Web, etc. in French and Japanese[18]. To extend this, we are using the ”Cesselin” Japanese-French dictionary[19] and the ”Tangorin” Japanese-English dictionary[20], in which we have annotated some headwords as potential CQs, according to originally given indications[21]. For the Type (b) one, it’s laborious to pin down phrasemes[22] in row data.
Eg.
une poignée de sable (a handful of sand), une pointe d’ironie (a touch of irony), un pouce de terre (a handful of soil).

French and English phrasemes are, however in many cases, composed of “Number + Noun + preposition (de, of) + Noun without article”.

The Type (b) CQs in the Phrase Book II have been collected from a parallel corpus according to the frequency of polylexical expressions, by using a software that can produce a list of keywords in context [23]. We have filtered the collected data as CQs by checking them with the UWs in the dictionary.

5 Specification of classifiers/quantifiers dictionary

We anticipate that our CQs dictionary will include about 8000 entries for each language according to manual count by 1% (8269 entries) random sampling from the Cesselin dictionary (its total number of entries is 826970).

At present, our CQs dictionary contains 3000 entries. The specification (microstructure) of its entries is as follows:

---

[16] Tori Bank is a sentence corpus which has developed at Tottori Unversity in Japan in 2007. http://unicorn.ike.tottori-u.ac.jp/toribank/about_toribank.html

[17] Cassell’s French-English, English-French dictionary: with appendices of proper names, French coins, weights, and measures with conversion tables.

[18] At present, the total number of registered entries is about 2000 for the Type (a) CQs and 1000 for Type (b) CQs, and it is becoming larger day by day.)

[19] The Cesselin is a printed dictionary published in 1939 and 1957 in Japan. It has been reprocessed into a numeric version equipped with a search engine by Mathieu Mangeot-Nagata in 2015 [Mangeot-Nagata (2016)]: https://jibiki.fr).

[20] http://tangorin.com/

[21] Eg. ken (軒) in the Cesselin (English translations have been added by us.)

ken (軒) n.m. Avant-toit, f. Maison. spé: s’emploie pour compter les maisons (special: used to count houses).
十二軒 (Jyû ni ken, 12 houses) douze maisons, 二 軒目です (C’est la deuxième maison, It’s the second house)

けん ken 軒 in the Tangorin dictionary:
suffix / counter:
1. counter for buildings (esp. houses)
彼女は鳥かごを軒からつるした。 She hung the cage from the eaves.
彼の叔父は家を十軒も持っている。 His uncle owns no fewer than ten houses.

[22] By ”phraseme” we mean a set phrase, an idiomatic phrase, a polylexical expression, etc.

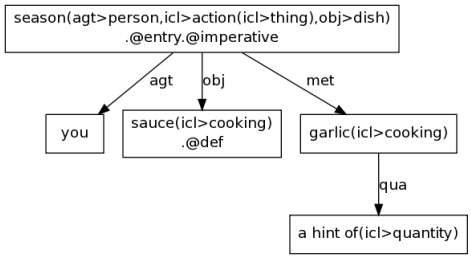
[23] http://en.wikipedia.org/wiki/Sketch_Engine

Table 4: Type (b) CQs in Phrase Book II : “pointe”

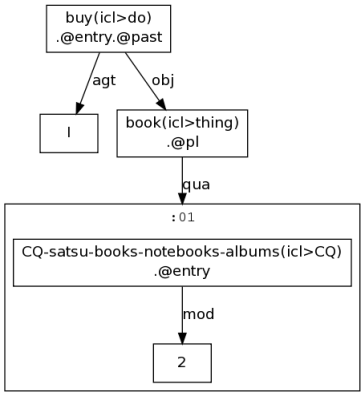
French word	examples	Source	Japanese translation	English translation
Pointe	une pointe d’ironie mal placée	J.L.Carré	場違いの皮肉をちくりと	the tip of , a hint of, a note of, a trace of
	relever la sauce avec une pointe d’ail	Livre de cuisine	ソースにニンニクを ちょっときかせる	pick up the sauce with a hint of garlic
	avec une pointe d’agacement dans la voix	T.Jonquet	声にすこし苦しみを にじませて	with a hint of irritation in the voice

Table 5: KWIC of “pointe” from Sketch Engine

doc#357	qui marque le déclin définitif de cette	pointe	de poussée et de sécrétions des hormones
doc#397	la sierra Pacaraima, qui constituent une	pointe	avancée du Sertao brésilien. En janvier
doc#457	de nouveauté, un soupçon de douceur, une	pointe	d’exotisme : commence par te mettre dans
doc#517	Tafer ne sont capables d’évoluer seuls en	pointe	. Arles - Marseille En concédant une



(a) Possible UNL-graph for ”Season the sauce with a hint of garlic.”



(b) UNL-graph for ”２冊の本を買いました”

Figure 2: Two UNL-graphs representing sentences containing CQs

Table 6: Description of “pointe”

items	description for ”pointe”
1.Identification number	XX
2. Keywords and class	pointe (n.)
3. English sentence	Season the sauce with a hint of garlic
4. French sentence	relever la sauce avec une pointe d’ail
5. Japanese sentence	ソースにニンニクをちょっときかせる
6. Source	Royal
7. UNL annotation	agt(season(agt>person, obj>dish, icl>action>thing).@entry.@imperative, you) obj(season(agt>person, obj>dish, icl>action>thing).@entry.@imperative, sauce(icl>cooking).@def) met(season(agt>person, obj>dish, icl>action>thing).@entry.@imperative, garlic(icl>cooking)) qua(garlic(icl>cooking), a hint of(icl>quantity))

Perspectives and Conclusion

We have studied the methodology for phraseology treatment on MT systems, while developing a French-Japanese-English parallel corpus and have known deeper linguistic analysis [Petit (2004), Gouverneur (2005)] is necessary for CQs dictionary description.

The corpus will be made freely accessible, so that software developers can use it. It should also be helpful for learners of languages, because it covers lexico-semantic information which cannot yet be found in any bilingual dictionary. We intend to produce a tool bilingual sentence-aligned corpus processing tool that will show corresponding (chunks of) words between 2 languages are shown on demand by character blinking or where the meaning of nouns or verbs in a sentence is shown without any ambiguity by interpreting UNL annotations. A prototype has been already presented by a Ph.D student in his thesis [Chenon (2005)].

References

Bobaljik, J. D. (2001). *Floating Quantifiers : Handle with care*. Mouton, France.

Boitet, C. and Tomokiyo, M. (1995). Ambiguity and ambiguity labelling : towards ambiguity data bases. In Mitkov, R., editor, *Proc. of RANLP-95 (Recent Advances in Natural Language Processing)*, Bulgaria.

Boitet, C. and Tomokiyo, M. (1996). On the formal definition of ambiguity and related concepts, leading to an ambiguity-labelling scheme. In Blanc, É. and Boitet, C., editors, *Actes de l’atelier post-COLING Multimedia Interactive Disambiguation / Désamguïsation Interactive Multimedia (MIDDIM-96)*, Le Col de Porte, France. MIDDIM-96 Post-COLING Seminar, GETA.

Chenon, C. (2005). *Vers une meilleure utilisabilité des mémoires de traductions, fondée sur un alignement sous-phrastique*. PhD thesis, Université Joseph Fourier, Grenoble.

Table 7: Description of 冊

items	description for ”冊”
1.Identification number	XX
2. Keywords and class	satsu(CQ-books, notebooks, albums)
3. English sentence	I bought 2 books.
4. French sentence	J'ai acheté 2 livres.
5. Japanese sentence	”2冊の本を買いました。”
6. Source	Royal
7. UNL annotation	agt(buy(icl>do).@entry.@past, I) obj(buy(icl>do).@entry.@past, book(icl>thing).@pl) qua(book(icl>thing).@pl, :01) mod:01(CQ-satsu-books-notebooks-albums(icl>CQ).@entry.@eld, 2)

Gouverneur, C. (2005). The phraseological patterns of high-frequency verbs in advanced English for general purposes. In *Proc. Sixth International Conference on Teaching and Language Corpora (TaLC6)*, pages 159–163, Louvain-La-Neuve. Université Catholique de Louvain (UCL).

Mangeot-Nagata, M. (2016). Collaborative construction of a good quality, broad coverage and copyright-free Japanese-French dictionary. *International Journal of Lexicography*.

Mari, A. (2011). *Quantificateurs polysémiques*. Mémoire de HDR (habilitation à diriger des recherches), Université IV Paris-Sorbonne.

Miyagawa, S. (1989). Structure and case marking in Japanese. New York.

Petit, G. (2004). La polysémie des séquences polylexicales, syntaxe et sémantique. *HAL*, 1(5):91–114.

Sérasset, G. and Boitet, C. (1999). UNL-French deconversion as transfer from an interlingua with possible quality enhancement through offline human interaction. In *Proc. Machine Translation Summit VII*, Singapore.

Tomokiyo, M. and Axtmeyer, M. (1996). Experiments in ambiguity labelling of dialogue transcriptions. In Boitet, C., editor, *Proc. MIDDIM-96 Post-COLING Seminar*, Le Sappey en Chartreuse. GETA.

Tomokiyo, M. and Boitet, C. (2016). Corpus and dictionary development for classifiers/quantifiers towards a French-Japanese Machine Translation. In *Proc. COLING-2016 CogALex workshop*, Osaka. ACL.

Uchida, H., Zhu, M., and Della Senta, T. G. (2006). *Universal Networking Language (UNL)*. UNDL Foundation, Japan.

Wisniewski, G., Singhand, A. K., Segal, N., and Yvon, F. (2013). Un corpus d'erreurs de traduction. In *Proc. of TALN-2013*, Les Sables d'Olonne, France. ATALA.

Annex「鳥バンク」

(examples from the Tori-Bank)

Eg.「塁 (rui, base)」,「寸 (sun, approx. 3.03 cm)」

AC00046100 P11:二塁走者の生還を許し:VP@28:allowing the runner to score from second:VP

AC00046100 P4:一塁へ悪投し、:VP@7:threw wild to first:VP

AC01599600 C6:一寸先も見え:CL@27:we could not see an inch ahea:CL

AC01599600 P6:一寸先も見え:VP@40:see an inch ahead:VP

Neural Machine Translation Model with a Large Vocabulary Selected by Branching Entropy

Zi Long
Ryuichiro Kimura
Takehito Utsuro
Grad. Sc. Sys. & Inf. Eng., University of Tsukuba, tsukuba, 305-8573, Japan

Tomoharu Mitsuhashi
Japan Patent Information Organization, 4-1-7, Tokyo, Koto-ku, Tokyo, 135-0016, Japan

Mikio Yamamoto
Grad. Sc. Sys. & Inf. Eng., University of Tsukuba, tsukuba, 305-8573, Japan

Abstract

Neural machine translation (NMT), a new approach to machine translation, has achieved promising results comparable to those of traditional approaches such as statistical machine translation (SMT). Despite its recent success, NMT cannot handle a larger vocabulary because the training complexity and decoding complexity proportionally increase with the number of target words. This problem becomes even more serious when translating patent documents, which contain many technical terms that are observed infrequently. In this paper, we propose to select phrases that contain out-of-vocabulary words using the statistical approach of branching entropy. This allows the proposed NMT system to be applied to a translation task of any language pair without any language-specific knowledge about technical term identification. The selected phrases are then replaced with tokens during training and post-translated by the phrase translation table of SMT. Evaluation on Japanese-to-Chinese, Chinese-to-Japanese, Japanese-to-English and English-to-Japanese patent sentence translation proved the effectiveness of phrases selected with branching entropy, where the proposed NMT model achieves a substantial improvement over a baseline NMT model without our proposed technique. Moreover, the number of translation errors of under-translation by the baseline NMT model without our proposed technique reduces to around half by the proposed NMT model.

1 Introduction

Neural machine translation (NMT), a new approach to solving machine translation, has achieved promising results (Bahdanau et al., 2015; Cho et al., 2014; Jean et al., 2014; Kalchbrenner and Blunsom, 2013; Luong et al., 2015a,b; Sutskever et al., 2014). An NMT system builds a simple large neural network that reads the entire input source sentence and generates an output translation. The entire neural network is jointly trained to maximize the conditional probability of the correct translation of a source sentence with a bilingual corpus. Although NMT offers many advantages over traditional phrase-based approaches, such as a small memory footprint and simple decoder implementation, conventional NMT is limited when it comes to larger vocabu-

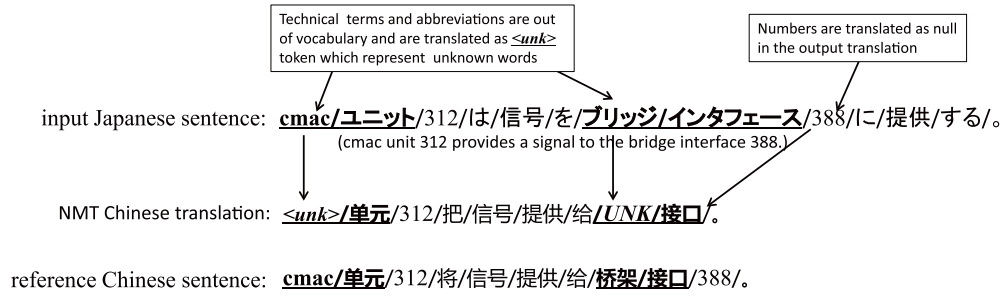


Figure 1: Example of translation errors when translating patent sentences with technical terms using NMT

laries. This is because the training complexity and decoding complexity proportionally increase with the number of target words. Words that are out of vocabulary are represented by a single "$\langle unk \rangle$" token in translations, as illustrated in Figure 1. The problem becomes more serious when translating patent documents, which contain several newly introduced technical terms.

There have been a number of related studies that address the vocabulary limitation of NMT systems. Jean et al. (2014) provided an efficient approximation to the softmax function to accommodate a very large vocabulary in an NMT system. Luong et al. (2015b) proposed annotating the occurrences of the out-of-vocabulary token in the target sentence with positional information to track its alignments, after which they replace the tokens with their translations using simple word dictionary lookup or identity copy. Li et al. (2016) proposed replacing out-of-vocabulary words with similar in-vocabulary words based on a similarity model learnt from monolingual data. Sennrich et al. (2016) introduced an effective approach based on encoding rare and out-of-vocabulary words as sequences of subword units. Luong and Manning (2016) provided a character-level and word-level hybrid NMT model to achieve an open vocabulary, and Costa-Jussà and Fonollosa (2016) proposed an NMT system that uses character-based embeddings.

However, these previous approaches have limitations when translating patent sentences. This is because their methods only focus on addressing the problem of out-of-vocabulary words even though the words are parts of technical terms. It is obvious that a technical term should be considered as one word that comprises components that always have different meanings and translations when they are used alone. An example is shown in Figure 1, where the Japanese word "ブリッジ"(bridge) should be translated to Chinese word "桥架" when included in technical term "bridge interface"; however, it is always translated as "桥".

To address this problem, Long et al. (2016) proposed extracting compound nouns as technical terms and replacing them with tokens. These compound nouns then are post-translated with the phrase translation table of the statistical machine translation (SMT) system. However, in their work on Japanese-to-Chinese patent translation, Japanese compound nouns are identified using several heuristic rules that use specific linguistic knowledge based on part-of-speech tags of morphological analysis of Japanese language, and thus, the NMT system has limited application to the translation task of other language pairs. In this paper, based on the approach of training an NMT model on a bilingual corpus wherein technical term pairs are replaced with tokens as in Long et al. (2016), we aim to select phrase pairs using the statistical approach of branching entropy; this allows the proposed technique to be applied to the translation task on any language pair without needing specific language knowledge to formulate the rules for technical term identification. Based on the results of our experiments on many pairs of languages: Japanese-to-Chinese, Chinese-to-Japanese, Japanese-to-English and English-to-Japanese, the

proposed NMT model achieves a substantial improvement over a baseline NMT model without our proposed technique. Our proposed NMT model achieves an improvement of 1.2 BLEU points over a baseline NMT model when translating Japanese sentences into Chinese, and an improvement of 1.7 BLEU points when translating Chinese sentences into Japanese. Our proposed NMT model achieves an improvement of 1.1 BLEU points over a baseline NMT model when translating Japanese sentences into English, and an improvement of 1.4 BLEU points when translating English sentences into Japanese. Moreover, the number of translation error of under-translations[1] by the the baseline NMT model without our proposed technique reduces to around half by the proposed NMT model.

2 Neural Machine Translation

NMT uses a single neural network trained jointly to maximize the translation performance (Bahdanau et al., 2015; Cho et al., 2014; Kalchbrenner and Blunsom, 2013; Luong et al., 2015a; Sutskever et al., 2014). Given a source sentence $\boldsymbol{x} = (x_1, \ldots, x_N)$ and target sentence $\boldsymbol{y} = (y_1, \ldots, y_M)$, an NMT model uses a neural network to parameterize the conditional distributions

$$p(y_z \mid y_{<z}, \boldsymbol{x})$$

for $1 \leq z \leq M$. Consequently, it becomes possible to compute and maximize the log probability of the target sentence given the source sentence as

$$\log p(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{l=1}^{M} \log p(y_z | y_{<z}, \boldsymbol{x})$$

In this paper, we use an NMT model similar to that used by Bahdanau et al. (2015), which consists of an encoder of a bidirectional long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and another LSTM as decoder. In the model of Bahdanau et al. (2015), the encoder consists of forward and backward LSTMs. The forward LSTM reads the source sentence as it is ordered (from x_1 to x_N) and calculates a sequence of forward hidden states, while the backward LSTM reads the source sentence in the reverse order (from x_N to x_1) , resulting in a sequence of backward hidden states. The decoder then predicts target words using not only a recurrent hidden state and the previously predicted word but also a context vector as followings:

$$p(y_z \mid y_{<z}, \boldsymbol{x}) = g(y_{z-1}, s_{z-1}, c_z)$$

where s_{z-1} is an LSTM hidden state of decoder, and c_z is a context vector computed from both of the forward hidden states and backward hidden states, for $1 \leq z \leq M$.

3 Phrase Pair Selection using Branching Entropy

Branching entropy has been applied to the procedure of text segmentation (e.g., (Jin and Tanaka-Ishii, 2006)) and key phrases extraction (e.g., (Chen et al., 2010)). In this work, we use the left/right branching entropy to detect the boundaries of phrases, and thus select phrase pairs automatically.

[1] It is known that NMT models tend to have the problem of the under-translation. Tu et al. (2016) proposed coverage-based NMT which considers the problem of the under-translation.

3.1 Branching Entropy

The left branching entropy and right branching entropy of a phrase $\boldsymbol{w}$ are respectively defined as

$$H_l(\boldsymbol{w}) = -\sum_{v \in V_l^{\boldsymbol{w}}} p_l(v) \log_2 p_l(v)$$

$$H_r(\boldsymbol{w}) = -\sum_{v \in V_r^{\boldsymbol{w}}} p_r(v) \log_2 p_r(v)$$

where $\boldsymbol{w}$ is the phrase of interest (e.g., "ブリッジ/インターフェース" in the Japanese sentence shown in Figure 1, which means "bridge interface"), $V_l^{\boldsymbol{w}}$ is a set of words that are adjacent to the left of $\boldsymbol{w}$ (e.g., "を" in Figure 1, which is a Japanese particle) and $V_r^{\boldsymbol{w}}$ is a set of words that are adjacent to the right of $\boldsymbol{w}$ (e.g., "388" in Figure 1). The probabilities $p_l(v)$ and $p_r(v)$ are respectively computed as

$$p_l(v) = \frac{f_{v,\boldsymbol{w}}}{f_{\boldsymbol{w}}} \qquad p_r(v) = \frac{f_{\boldsymbol{w},v}}{f_{\boldsymbol{w}}}$$

where $f_{\boldsymbol{w}}$ is the frequency count of phrase $\boldsymbol{w}$, and $f_{v,\boldsymbol{w}}$ and $f_{\boldsymbol{w},v}$ are the frequency counts of sequence "v,$\boldsymbol{w}$" and sequence "$\boldsymbol{w}$,v" respectively. According to the definition of branching entropy, when a phrase $\boldsymbol{w}$ is a technical term that is always used as a compound word, both its left branching entropy $H_l(\boldsymbol{w})$ and right branching entropy $H_r(\boldsymbol{w})$ have high values because many different words, such as particles and numbers, can be adjacent to the phrase. However, the left/right branching entropy of substrings of $\boldsymbol{w}$ have low values because words contained in $\boldsymbol{w}$ are always adjacent to each other.

3.2 Selecting Phrase Pairs

Given a parallel sentence pair $\langle S_s, S_t \rangle$, all n-grams phrases of source sentence S_s and target sentence S_t are extracted and aligned using phrase translation table and word alignment of SMT according to the approaches described in Long et al. (2016). Next, phrase translation pair $\langle t_s, t_t \rangle$ obtained from $\langle S_s, S_t \rangle$ that satisfies all the following conditions is selected as a phrase pair and is extracted:

(1) Either t_s or t_t contains at least one out-of-vocabulary word.[2]

(2) Neither t_s nor t_t contains predetermined stop words.

(3) Entropies $H_l(t_s)$, $H_l(t_t)$, $H_r(t_s)$ and $H_r(t_t)$ are larger than a lower bound, while the left/right branching entropy of the substrings of t_s and t_t are lower than or equal to the lower bound.

Here, the maximum length of a phrase as well as the lower bound of the branching entropy are tuned with the validation set.[3] All the selected source-target phrase pairs are then used in the

[2] One of the major focus of this paper is the comparison between the proposed method and Luong et al. (2015b). Since Luong et al. (2015b) proposed to pre-process and post-translate only out-of-vocabulary words, we focus only on compound terms which include at least one out-of-vocabulary words.

[3] Throughout the evaluations on patent translation of both language pairs of Japanese-Chinese and Japanese-English, the maximum length of the extracted phrases is tuned as 7. The lower bounds of the branching entropy are tuned as 5 for patent translation of the language pair of Japanese-Chinese, and 8 for patent translation of the language pair of Japanese-English. We also tune the number of stop words using the validation set, and use the 200 most-frequent Japanese morphemes and Chinese words as stop words for the language pair of Japanese-Chinese, use the 100 most-frequent Japanese morphemes and English words as stop words for the language pair of Japanese-English.

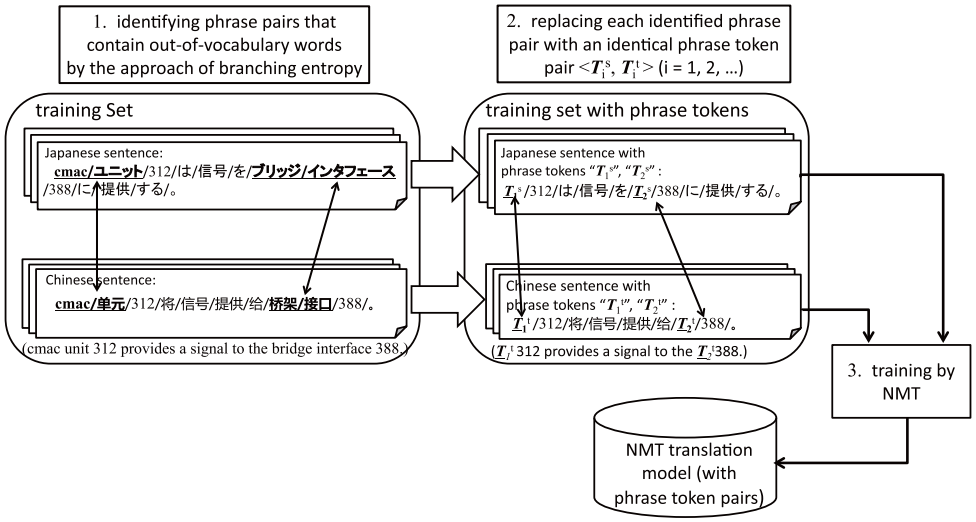


Figure 2: NMT training after replacing phrase pairs with token pairs $\langle T_i^s, T_i^t \rangle$ $(i = 1, 2, \ldots)$

next section as phrase pairs.[4]

4 NMT with a Large Phrase Vocabulary

In this work, the NMT model is trained on a bilingual corpus in which phrase pairs are replaced with tokens. The NMT system is then used as a decoder to translate the source sentences and replace the tokens with phrases translated using SMT.

4.1 NMT Training after Replacing Phrase Pairs with Tokens

Figure 2 illustrates the procedure for training the model with parallel patent sentence pairs in which phrase pairs are replaced with phrase token pairs $\langle T_1^s, T_1^t \rangle$, $\langle T_2^s, T_2^t \rangle$, and so on.

In the step 1 of Figure 2, source-target phrase pairs that contain at least one out-of-vocabulary word are selected from the training set using the branching entropy approach described in Section 3.2. As shown in the step 2 of Figure 2, in each of the parallel patent sentence pairs, occurrences of phrase pairs $\langle t_1^s, t_1^t \rangle$, $\langle t_2^s, t_2^t \rangle$, ..., $\langle t_k^s, t_k^t \rangle$ are then replaced with token pairs $\langle T_1^s, T_1^t \rangle$, $\langle T_2^s, T_2^t \rangle$, ..., $\langle T_k^s, T_k^t \rangle$. Phrase pairs $\langle t_1^s, t_1^t \rangle$, $\langle t_2^s, t_2^t \rangle$, ..., $\langle t_k^s, t_k^t \rangle$ are numbered in the order of occurrence of the source phrases t_1^s $(i = 1, 2, \ldots, k)$ in each source sentence S_s. Here note that in all the parallel sentence pairs $\langle S_s, S_t \rangle$, the tokens pairs $\langle T_1^s, T_1^t \rangle$, $\langle T_2^s, T_2^t \rangle$, ... that are identical throughout all the parallel sentence pairs are used in this procedure. Therefore, for example, in all the source patent sentences S_s, the phrase t_1^s which appears earlier than other phrases in S_s is replaced with T_1^s. We then train the NMT model on a bilingual corpus, in which the phrase pairs are replaced by token pairs $\langle T_i^s, T_i^t \rangle$ $(i = 1, 2, \ldots)$, and obtain an NMT model in which the phrases are represented as tokens.[5]

[4] We sampled 200 Japanese-Chinese sentence pairs, manually annotated compounds and evaluated the approach of phrase extraction with the branching entropy. Based on the result, (a) 25% of them are correct, (b) 20% subsume correct compounds as their substrings, (c) 18% are substrings of correct compounds, (d) 22% subsume substrings of correct compounds but other than (b) nor (c), and (e) the remaining 15% are error strings such as functional compounds and fragmental strings consisting of numerical expressions.

[5] We treat the NMT system as a black box, and the strategy we present in this paper could be applied to any NMT system (Bahdanau et al., 2015; Cho et al., 2014; Kalchbrenner and Blunsom, 2013; Luong et al., 2015a; Sutskever et al.,

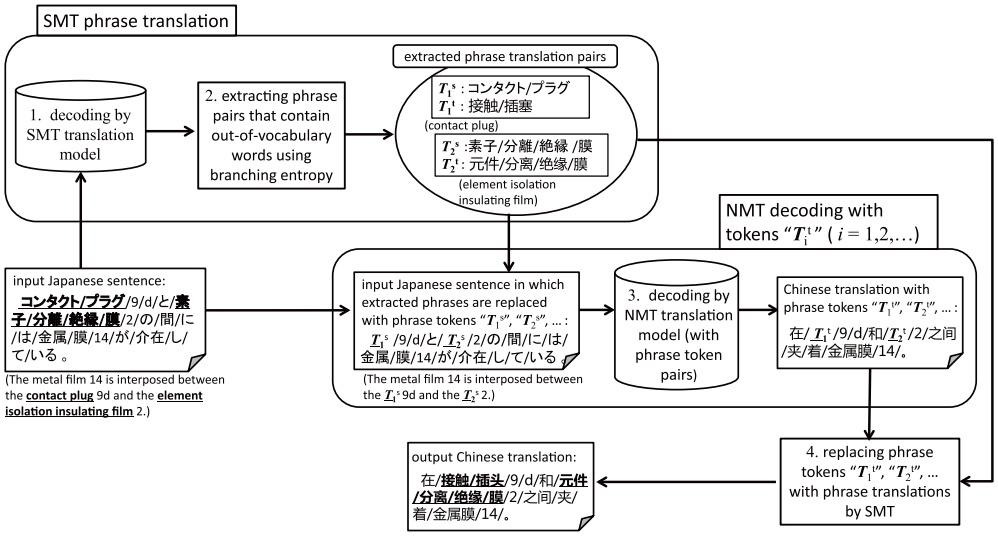


Figure 3: NMT decoding with tokens "T_i^s" ($i = 1, 2, \ldots$) and the SMT phrase translation

4.2 NMT Decoding and SMT Phrase Translation

Figure 3 illustrates the procedure for producing target translations by decoding the input source sentence using the method proposed in this paper.

In the step 1 of Figure 3, when given an input source sentence, we first generate its translation by decoding of SMT translation model. Next, as shown in the step 2 of Figure 3, we automatically extract the phrase pairs by branching entropy according to the procedure of Section 3.2, where the input sentence and its SMT translation are considered as a pair of parallel sentence. Phrase pairs that contains at least one out-of-vocabulary word are extracted and are replaced with phrase token pairs $\langle T_i^s, T_i^t \rangle$ ($i = 1, 2, \ldots$). Consequently, we have an input sentence in which the tokens "T_i^s" ($i = 1, 2, \ldots$) represent the positions of the phrases and a list of SMT phrase translations of extracted Japanese phrases. Next, as shown in the step 3 of Figure 3, the source Japanese sentence with tokens is translated using the NMT model trained according to the procedure described in Section 4.1. Finally, in the step 4, we replace the tokens "T_i^t" ($i = 1, 2, \ldots$) of the target sentence translation with the phrase translations of the SMT.

5 Evaluation

5.1 Patent Documents

Japanese-Chinese parallel patent documents were collected from the Japanese patent documents published by the Japanese Patent Office (JPO) during 2004-2012 and the Chinese patent documents published by the State Intellectual Property Office of the People's Republic of China (SIPO) during 2005-2010. From the collected documents, we extracted 312,492 patent families, and the method of Utiyama and Isahara (2007) was applied[6] to the text of the extracted patent families to align the Japanese and Chinese sentences. The Japanese sentences were segmented into a sequence of morphemes using the Japanese morphological analyzer MeCab [7] with

2014).

[6]Herein, we used a Japanese-Chinese translation lexicon comprising around 170,000 Chinese entries.

[7]http://mecab.sourceforge.net/

Table 1: Statistics of datasets

	training set	validation set	test set
Japanese-Chinese	2,877,178	1,000	1,000
Japanese-English	1,167,198	1,000	1,000

Table 2: Automatic evaluation results (BLEU)

System	ja → ch	ch → ja	ja → en	en → ja
Baseline SMT (Koehn et al., 2007)	52.5	57.1	32.3	32.1
Baseline NMT	56.5	62.5	39.9	41.5
NMT with PosUnk model (Luong et al., 2015b)	56.9	62.9	40.1	41.9
NMT with phrase translation by SMT (phrase pairs selected with branching entropy)	**57.7**	**64.2**	**40.3**	**42.9**

the morpheme lexicon IPAdic,[8] and the Chinese sentences were segmented into a sequence of words using the Chinese morphological analyzer Stanford Word Segment (Tseng et al., 2005) trained using the Chinese Penn Treebank. In this study, Japanese-Chinese parallel patent sentence pairs were ordered in descending order of sentence-alignment score and we used the topmost 2.8M pairs, whose Japanese sentences contain fewer than 40 morphemes and Chinese sentences contain fewer than 40 words.[9]

Japanese-English patent documents are provided in the NTCIR-7 workshop (Fujii et al., 2008), which are collected from the 10 years of unexamined Japanese patent applications published by the Japanese Patent Office (JPO) and the 10 years patent grant data published by the U.S. Patent & Trademark Office (USPTO) in 1993-2000. The numbers of documents are approximately 3,500,000 for Japanese and 1,300,000 for English. From these document sets, patent families are automatically extracted and the fields of "Background of the Invention" and "Detailed Description of the Preferred Embodiments" are selected. Then, the method of Utiyama and Isahara (2007) is applied to the text of those fields, and Japanese and English sentences are aligned. The Japanese sentences were segmented into a sequence of morphemes using the Japanese morphological analyzer MeCab with the morpheme lexicon IPAdic. Similar to the case of Japanese-Chinese patent documents, in this study, out of the provided 1.8M Japanese-English parallel sentences, 1.1M parallel sentences whose Japanese sentences contain fewer than 40 morphemes and English sentences contain fewer than 40 words are used.

5.2 Training and Test Sets

We evaluated the effectiveness of the proposed NMT model at translating parallel patent sentences described in Section 5.1. Among the selected parallel sentence pairs, we randomly extracted 1,000 sentence pairs for the test set and 1,000 sentence pairs for the validation set; the remaining sentence pairs were used for the training set. Table 1 shows statistics of the datasets.

According to the procedure of Section 3.2, from the Japanese-Chinese sentence pairs of the training set, we collected 426,551 occurrences of Japanese-Chinese phrase pairs, which

[8] `http://sourceforge.jp/projects/ipadic/`

[9] It is expected that the proposed NMT model can improve the baseline NMT without the proposed technique when translating longer sentences that contain more than 40 morphemes / words. It is because the approach of replacing phrases with tokens also shortens the input sentences, expected to contribute to solving the weakness of NMT model when translating long sentences.

Table 3: Human evaluation results of pairwise evaluation (the score ranges from −100 to 100)

System	ja → ch	ch → ja	ja → en	en → ja
Baseline NMT	-	-	-	-
NMT with PosUnk model (Luong et al., 2015b)	9	10.5	8	6.5
NMT with phrase translation by SMT (phrase pairs selected with branching entropy)	**14.5**	**17**	**11.5**	**15.5**

are 254,794 types of phrase pairs with 171,757 unique types of Japanese phrases and 129,071 unique types of Chinese phrases. Within the total 1,000 Japanese patent sentences in the Japanese-Chinese test set, 121 occurrences of Japanese phrases were extracted, which correspond to 120 types. With the total 1,000 Chinese patent sentences in the Japanese-Chinese test set, 130 occurrences of Chinese phrases were extracted, which correspond to 130 types.

From the Japanese-English sentence pairs of the training set, we collected 70,943 occurrences of Japanese-English phrase pairs, which are 61,017 types of phrase pairs with unique 57,675 types of Japanese phrases and 58,549 unique types of English phrases. Within the total 1,000 Japanese patent sentences in the Japanese-English test set, 59 occurrences of Japanese phrases were extracted, which correspond to 59 types. With the total 1,000 English patent sentences in the Japanese-English test set, 61 occurrences of English phrases were extracted, which correspond to 61 types.

5.3 Training Details

For the training of the SMT model, including the word alignment and the phrase translation table, we used Moses (Koehn et al., 2007), a toolkit for phrase-based SMT models. We trained the SMT model on the training set and tuned it with the validation set.

For the training of the NMT model, our training procedure and hyperparameter choices were similar to those of Bahdanau et al. (2015). The encoder consists of forward and backward deep LSTM neural networks each consisting of three layers, with 512 cells in each layer. The decoder is a three-layer deep LSTM with 512 cells in each layer. Both the source vocabulary and the target vocabulary are limited to the 40K most-frequently used morphemes / words in the training set. The size of the word embedding was set to 512. We ensured that all sentences in a minibatch were roughly the same length. Further training details are given below: (1) We set the size of a minibatch to 128. (2) All of the LSTM's parameter were initialized with a uniform distribution ranging between -0.06 and 0.06. (3) We used the stochastic gradient descent, beginning at a fixed learning rate of 1. We trained our model for a total of 10 epochs, and we began to halve the learning rate every epoch after the first seven epochs. (4) Similar to Sutskever et al. (2014), we rescaled the normalized gradient to ensure that its norm does not exceed 5. We trained the NMT model on the training set. The training time was around two days when using the described parameters on a 1-GPU machine.

We compute the branching entropy using the frequency statistics from the training set.

5.4 Evaluation Results

In this work, we calculated automatic evaluation scores for the translation results using a popular metrics called BLEU (Papineni et al., 2002). As shown in Table 2, we report the evaluation scores, using the translations by Moses (Koehn et al., 2007) as the baseline SMT and the scores using the translations produced by the baseline NMT system without our proposed approach as the baseline NMT. As shown in Table 2, the BLEU score obtained by the proposed NMT

Table 4: Human evaluation results of JPO adequacy evaluation (the score ranges from 1 to 5)

System	ja → ch	ch → ja	ja → en	en → ja
Baseline SMT (Koehn et al., 2007)	3.5	3.7	3.1	3.2
Baseline NMT	4.2	4.3	3.9	4.1
NMT with PosUnk model (Luong et al., 2015b)	4.3	4.3	4.0	4.2
NMT with phrase translation by SMT (phrase pairs selected with branching entropy)	**4.5**	**4.6**	**4.1**	**4.4**

Table 5: Numbers of untranslated morphemes / words of input sentences (for the test set)

System	ja → ch	ch → ja	ja → en	en → ja
Baseline NMT	89	92	415	226
NMT with phrase translation by SMT (phrase pairs selected with branching entropy)	43	45	246	134

model is clearly higher than those of the baselines. Here, as described in Section 3, the lower bounds of branching entropy for phrase pair selection are tuned as 5 throughout the evaluation of language pair of Japanese-Chinese, and tuned as 8 throughout the evaluation of language pair of Japanese-English, respectively. When compared with the baseline SMT, the performance gains of the proposed system are approximately 5.2 BLEU points when translating Japanese into Chinese and 7.1 BLEU when translating Chinese into Japanese. When compared with the baseline SMT, the performance gains of the proposed system are approximately 10.0 BLEU points when translating Japanese into English and 10.8 BLEU when translating English into Japanese. When compared with the result of the baseline NMT, the proposed NMT model achieved performance gains of 1.2 BLEU points on the task of translating Japanese into Chinese and 1.7 BLEU points on the task of translating Chinese into Japanese. When compared with the result of the baseline NMT, the proposed NMT model achieved performance gains of 0.4 BLEU points on the task of translating Japanese into English and 1.4 BLEU points on the task of translating English into Japanese.

Furthermore, we quantitatively compared our study with the work of Luong et al. (2015b). Table 2 compares the NMT model with the PosUnk model, which is the best model proposed by Luong et al. (2015b). The proposed NMT model achieves performance gains of 0.8 BLEU points when translating Japanese into Chinese, and performance gains of 1.3 BLEU points when translating Chinese into Japanese. The proposed NMT model achieves performance gains of 0.2 BLEU points when translating Japanese into English, and performance gains of 1.0 BLEU points when translating English into Japanese.

We also compared our study with the work of Long et al. (2016). As reported in Long et al. (2017), when translating Japanese into Chinese, the BLEU of the NMT system of Long et al. (2016) in which all the selected compound nouns are replaced with tokens is 58.6, the BLEU of the NMT system in which only compound nouns that contain out-of-vocabulary words are selected and replaced with tokens is 57.4, while the BLEU of the proposed NMT system of this paper is 57.7. Out of all the selected compound nouns of Long et al. (2016), around 22% contain out-of-vocabulary words, of which around 36% share substrings with the phrases selected by branching entropy. The remaining 78% compound nouns do not contain out-of-vocabulary words and are considered to contribute to the improvement of BLEU points compared with the proposed method. Based on this analysis, as one of our important future work, we revise the

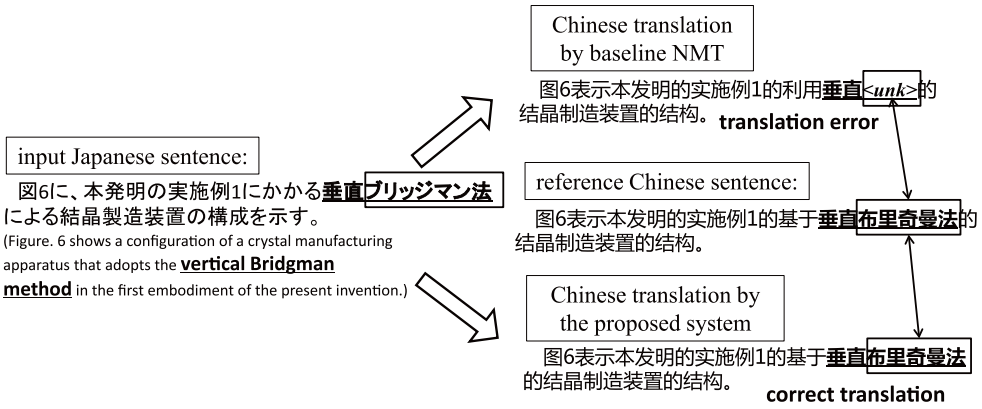


Figure 4: An example of correct translations produced by the proposed NMT model when addressing the problem of out-of-vocabulary words (Japanese-to-Chinese)

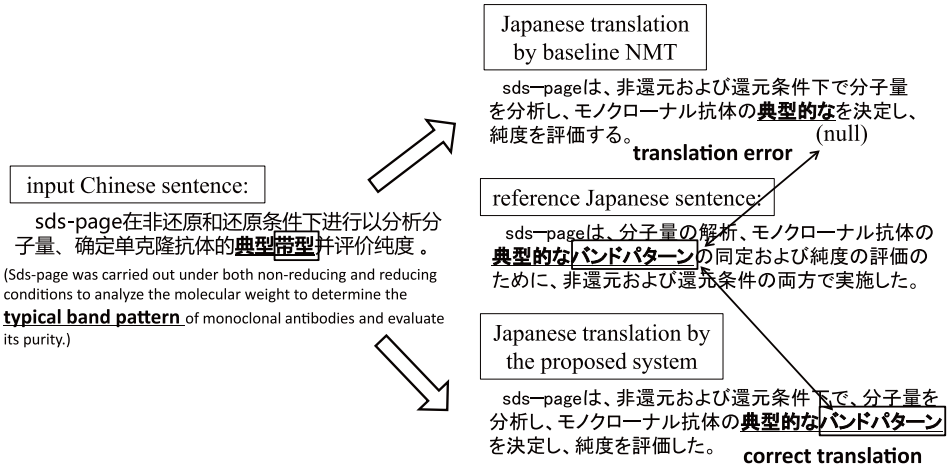


Figure 5: An example of correct translations produced by the proposed NMT model when addressing the problem of under-translation (Chinese-to-Japanese)

procedure in Section 3.2 of selecting phrases by branching entropy and then incorporate those in-vocabulary compound nouns into the set of the phrases selected by the branching entropy.

In this study, we also conducted two types of human evaluations according to the work of Nakazawa et al. (2015): pairwise evaluation and JPO adequacy evaluation. In the pairwise evaluation, we compared each translation produced by the baseline NMT with that produced by the proposed NMT model as well as the NMT model with PosUnk model, and judged which translation is better or whether they have comparable quality. The score of the pairwise evaluation is defined as below:

$$score = 100 \times \frac{W - L}{W + L + T}$$

where W, L, and T are the numbers of translations that are better than, worse than, and comparable to the baseline NMT, respectively. The score of pairwise evaluation ranges from -100 to 100. In the JPO adequacy evaluation, Chinese translations are evaluated according to the quality evaluation criterion for translated patent documents proposed by the Japanese Patent

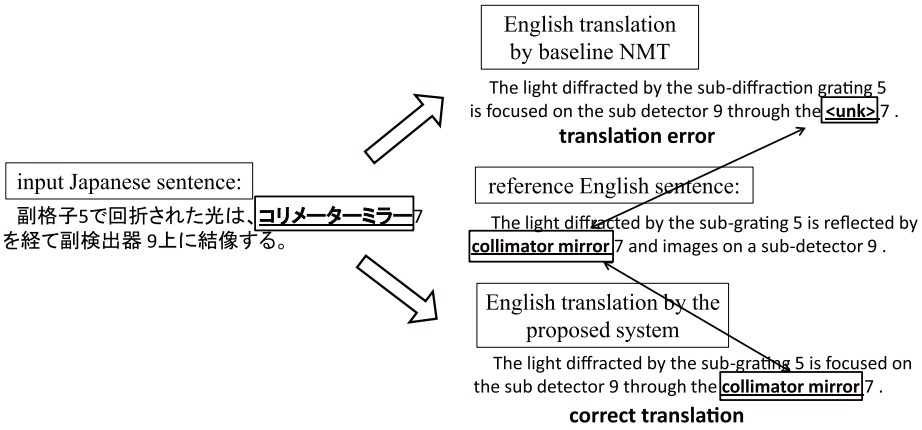


Figure 6: An example of correct translations produced by the proposed NMT model when addressing the problem of out-of-vocabulary words (Japanese-to-English)

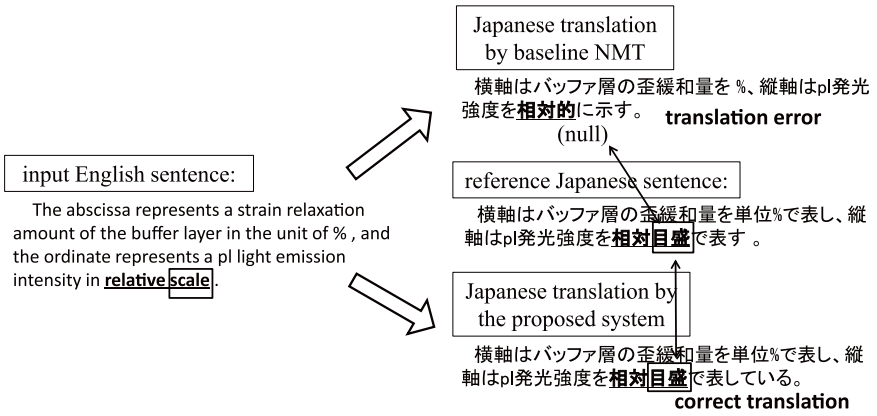


Figure 7: An example of correct translations produced by the proposed NMT model when addressing the problem of under-translation (English-to-Japanese)

Office (JPO).[10] The JPO adequacy criterion judges whether or not the technical factors and their relationships included in Japanese patent sentences are correctly translated into Chinese. The Chinese translations are then scored according to the percentage of correctly translated information, where a score of 5 means all of those information are translated correctly, while a score of 1 means that most of those information are not translated correctly. The score of the JPO adequacy evaluation is defined as the average over all the test sentences. In contrast to the study conducted by Nakazawa et al. (2015), we randomly selected 200 sentence pairs from the test set for human evaluation, and both human evaluations were conducted using only one judgement. Table 3 and Table 4 shows the results of the human evaluation for the baseline SMT, baseline NMT, NMT model with PosUnk model, and the proposed NMT model. We observe that the proposed model achieves the best performance for both the pairwise and JPO adequacy evaluations when we replace the tokens with SMT phrase translations after decoding the source sentence with the tokens.

[10] https://www.jpo.go.jp/shiryou/toushin/chousa/pdf/tokkyohonyaku_hyouka/01.pdf (in Japanese)

For the test set, we also counted the numbers of the untranslated words of input sentences. As shown in Table 5, the number of untranslated words by the baseline NMT reduced to around 50% in the cases of ja → ch and ch → ja by the proposed NMT model, and reduced to around 60% in the cases of ja → en and en → ja.[11] [12] This is mainly because part of untranslated source words are out-of-vocabulary, and thus are untranslated by the baseline NMT. The proposed system extracts those out-of-vocabulary words as a part of phrases and replaces those phrases with tokens before the decoding of NMT. Those phrases are then translated by SMT and inserted in the output translation, which ensures that those out-of-vocabulary words are translated.

Figure 4 compares an example of correct translation produced by the proposed system with one produced by the baseline NMT. In this example, the translation is a translation error because the Japanese word "ブリッジマン (Bridgman)" is an out-of-vocabulary word and is erroneously translated into the "$\langle unk \rangle$" token. The proposed NMT model correctly translated the Japanese sentence into Chinese, where the out-of-vocabulary word "ブリッジマン" is correctly selected by the approach of branching entropy as a part of the Japanese phrase "垂直ブリッジマン法 (vertical Bridgman method)". The selected Japanese phrase is then translated by the phrase translation table of SMT. Figure 5 shows another example of correct translation produced by the proposed system with one produced by the baseline NMT. As shown in Figure 5, the translation produced by baseline NMT is a translation error because the out-of-vocabulary Chinese word "带型 (band pattern)" is an untranslated word and its translation is not contained in the output translation of the baseline NMT. The proposed NMT model correctly translated the Chinese word into Japanese because the Chinese word "带型 (band pattern)"is selected as a part of Chinese phrase "典型带型(typical band pattern)" with branching entropy and then is translated by SMT. Moreover, Figure 6 and Figure 7 compare examples of correct translations produced by the proposed system with those produced by the baseline NMT when translating patent sentences in both directions of Japanese-to-English and English-to-Japanese.

6 Conclusion

This paper proposed selecting phrases that contain out-of-vocabulary words using the branching entropy. These selected phrases are then replaced with tokens and post-translated using an SMT phrase translation. Compared with the method of Long et al. (2016), the contribution of the proposed NMT model is that it can be used on any language pair without language-specific knowledge for technical terms selection. We observed that the proposed NMT model performed much better than the baseline NMT system in all of the language pairs: Japanese-to-Chinese/Chinese-to-Japanese and Japanese-to-English/English-to-Japanese. One of our important future tasks is to compare the translation performance of the proposed NMT model with that based on subword units (e.g. Sennrich et al. (2016)). Another future task is to improve the performance of the present study by incorporating the in-vocabulary non-compositional phrases, whose translations cannot be obtained by translating their constituent words. It is expected to achieve a better translation performance by translating those kinds of phrases using a phrase-based SMT instead of using NMT.

[11] Although we omit the detail of the evaluation results of untranslated words of the NMT model with PosUnk model (Luong et al., 2015b) in Table 5, the number of the untranslated words of the NMT model with PosUnk model is almost the same as that of the baseline NMT, which is much more than that of the proposed NMT model.

[12] Following the result of an additional evaluation where having approximately similar size of the training parallel sentences between the language pairs of Japanese-to-Chinese/Chinese-to-Japanese and Japanese-to-English/English-to-Japanese, we concluded that the primary reason why the numbers of untranslated morphemes / words tend to be much larger in the case of the language pair of Japanese-to-English/English-to-Japanese than in the case of the language pair of Japanese-to-Chinese/Chinese-to-Japanese is simply the matter of a language specific issue.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proc. 3rd ICLR*.

Chen, Y., Huang, Y., Kong, S., and Lee, L. (2010). Automatic key term extraction from spoken course lectures using branching entropy and prosodic/semantic features. In *Proc. 2010 IEEE SLT Workshop*, pages 265–270.

Cho, K., van Merriënboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. EMNLP*, pages 1724–1734.

Costa-Jussà, M. R. and Fonollosa, J. A. R. (2016). Character-based neural machine translation. In *Proc. 54th ACL*, pages 357–361.

Fujii, A., Utiyama, M., Yamamoto, M., and Utsuro, T. (2008). Toward the evaluation of machine translation using patent information. In *Proc. 8th AMTA*, pages 97–106.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jean, S., Cho, K., Bengio, Y., and Memisevic, R. (2014). On using very large target vocabulary for neural machine translation. In *Proc. 28th NIPS*, pages 1–10.

Jin, Z. and Tanaka-Ishii, K. (2006). Unsupervised segmentation of Chinese text by use of branching entropy. In *Proc. COLING/ACL 2006*, pages 428–435.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proc. EMNLP*, pages 1700–1709.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proc. 45th ACL, Companion Volume*, pages 177–180.

Li, X., Zhang, J., and Zong, C. (2016). Towards zero unknown word in neural machine translation. In *Proc. 25th IJCAI*, pages 2852–2858.

Long, Z., Utsuro, T., Mitsuhashi, T., and Yamamoto, M. (2016). Translation of patent sentences with a large vocabulary of technical terms using neural machine translation. In *Proc. 3rd WAT*, pages 47–57.

Long, Z., Utsuro, T., Mitsuhashi, T., and Yamamoto, M. (2017). Neural machine translation model with a large vocabulary selected by branching entropy. `https://arxiv.org/abs/1704.04520v4`. Online; accessed 24-July-2017.

Luong, M. and Manning, C. D. (2016). Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proc. 54th ACL*, pages 1054–1063.

Luong, M., Pham, H., and Manning, C. D. (2015a). Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*, pages 1412–1421.

Luong, M., Sutskever, I., Vinyals, O., Le, Q. V., and Zaremba, W. (2015b). Addressing the rare word problem in neural machine translation. In *Proc. 53rd ACL*, pages 11–19.

Nakazawa, T., Mino, H., Goto, I., Neubig, G., Kurohashi, S., and Sumita, E. (2015). Overview of the 2nd workshop on Asian translation. In *Proc. 2nd WAT*, pages 1–28.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proc. 40th ACL*, pages 311–318.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proc. 54th ACL*, pages 1715–1725.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural machine translation. In *Proc. 27th NIPS*, pages 3104–3112.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., and Manning, C. (2005). A conditional random field word segmenter for Sighan bakeoff 2005. In *Proc. 4th SIGHAN Workshop on Chinese Language Processing*, pages 168–171.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016). Modeling coverage for neural machine translation. In *Proc. ACL 2016*, pages 76–85.

Utiyama, M. and Isahara, H. (2007). A Japanese-English patent parallel corpus. In *Proc. MT Summit XI*, pages 475–482.

Usefulness of MT output for comprehension — an analysis from the point of view of linguistic intercomprehension

Kenneth Jordan Núñez kjordan@usj.es
Universidad San Jorge, E-50830 Villanueva del Gállego, Spain

Mikel L. Forcada mlf@ua.es
Universitat d'Alacant, E-03690 Sant Vicent del Raspeig, Spain

Esteve Clua esteve.clua@upf.edu
Universitat Pompeu Fabra, E-08018 Barcelona, Spain

Abstract

The present paper describes and presents ongoing research on machine translation (MT) and linguistic intercomprehension. One main goal, although not the only one, is to evaluate three machine translation (MT) systems —Systran, Google Translate and Apertium— through an analysis of the readers' ability to understand the output generated. We compare the usefulness of MT output for comprehension to that of non-native writing in the readers' L_1 and that of native writing in languages similar to their L_1. The methodology used is based on *cloze* tests and the experiments are carried out using English, French and Italian as source languages and Spanish as the target language. The subjects involved are native Spanish first-year-undergraduate students and final-year secondary-school students. All of them have only very elementary knowledge, or in some cases no knowledge at all, of English, French and Italian (that is, a level equal to or lower than CEFRL B1[1]). Although the results suggest that MT output resulting from translating from English and French into Spanish is similar to natively-written Italian texts or texts written in Spanish by non-native speakers in terms of usefulness, that depends quite often on the level of specialty but also on the field and on the MT system used.

1 Introduction

The aim of the study, which is part of a broader research plan relating machine translation (MT) and linguistic intercomprehehsion, is to assess and compare three MT systems when used for assimilation[2] or *gisting* – Systran, a hybrid system[3] that combines statistical or corpus-based MT and rule-based MT; Google Translate,[4] a statistical corpus-based MT system at the time of testing for the language pairs tested,[5] and the Apertium rule-based system.[6] The aim is achieved

[1] Common European Framework of Reference for Languages: Learning, Teaching, Assessment (`http://www.coe.int/t/dg4/linguistic/Source/Framework\_EN.pdf`)

[2] MT systems can be divided into two groups: those aimed at *assimilation* or *gisting*, which allow the user to understand the content of the text, and those aimed at *dissemination*, which helps to translate a text to be published.

[3] `http://www.systran.es`

[4] `http://translate.google.com`

[5] As of May 15, 2017, the English–Spanish system and the English–Italian systems are no longer statistical, but rather neural; the French–Spanish system is still statistical.

[6] `http://www.apertium.org`

by comparing the usefulness of their output to that of non-native writing in the reader's first language L_1 and to that of native writings (or MT output) in a closely related language. This will enable us to determine which of the three MT systems is the most useful for gisting, that is, which MT system results in the highest level of usefulness when reading a text originally written in a language unknown to the reader.

Linguistic intercomprehension, the ability to understand one foreign language based on knowledge about another language (Meissner, 2004, 34), is an ability that the readers of a language naturally have and use unconsciously but that they can also develop in order to understand messages in another language without being able to produce them themselves (Martín-Peris, 2011, 247). Linguistic intercomprehension —in this work, *reading* or *written* intercomprehension, in contrast to *listening* or *spoken* intercomprehension— leverages on the similarity or identity of word forms and structures (Martín-Peris, 2011, 276).

In recent decades, although only in Europe (within the framework of Euro-comprehension or European intercomprehension), a new discipline focused on research into study methods has been developed, which is aimed at the simultaneous studying or learning of multiple languages. This discipline is centred on reading comprehension —as well as listening comprehension in some cases— and seeks to save time and effort when learning languages from the same linguistic family (Clua, 2003). Some of the methods designed have been adapted to Romance-language learning, such as Eurom4 and Eurom5,[7] Galatea,[8] Miriadi,[9] or EuroComRom[10] (within EuroCom, whose name refers straightforwardly to Euro-comprehension). EuroComRom shows the reader how to obtain information from texts in other languages —even if those languages are completely unknown to the reader— through the so-called *seven sieves* (Klein and Stegmann, 2000): international vocabulary or internationalisms, pan-Romance vocabulary, sound correspondences, spelling and pronunciation, basic structure of Romance sentence patterns, common morpho-syntactic structures developed by the Pan-Romance community, and the transfer of EuroFixes[11] (Martín Peris et al., 2005).

The work in this paper explores the quantitative aspects of a line of research that aims at exploring to what extent the differences between native text and MT output are similar to the differences between languages within a language family or the differences between native text and non-natively written text, and, then, explores whether spontaneous intercomprehension strategies (Klein and Stegmann, 2000; Martín Peris et al., 2005) play a role in the processing of raw machine-translated text by readers.

2 Research questions and hypotheses

We aim to answer four research questions: RQ1) To what extent are MT output and a text written by a native speaker in a language L' from the same language family as the reader's first language L_1 similar to the reader in terms of usefulness?; RQ2) Could the MT output in a language L' in the same language family as the reader's first language L_1 be useful for comprehension when translating texts originally written in a language from a language family different from that of their first language L_1?; RQ3) Are MT output and a text written in the reader's first language by non-native speakers similar to the reader in terms of usefulness?; and, lastly, RQ4) Which MT system is the most useful for comprehension when translating a text originally written in a language that is completely unknown to the reader?

[7] http://www.eurom5.com

[8] http://galatea.u-grenoble3.fr

[9] https://www.miriadi.net/en

[10] http://www.eurocomresearch.net

[11] Eurofixes are lexical components used in word formation, such as prefixes and suffixes, and which are shared across European languages, Many of them are Latin- and Greek-based affixes (*pseudo-*, *-phobia*, *inter-*, etc.).

Regarding RQ1, one could argue that the relationship between the reader's L_1 (first language, *mother tongue*) and MT output into L_1 has similarities to that between L_1 and other languages in the family of L_1 (common international vocabulary, common morpho-syntactic structures), and that, therefore, the usefulness of MT output into L_1 is similar to that of the related language L'.

Regarding RQ2, by analysing the usefulness to Spanish (ES) native speakers of MT output when translating from English (EN) into Italian (IT), we can determine if MT could be used to read texts originally written in a language from a different language family when a certain language combination is not available; that is, if a speaker of (ES) wants to understand a text in EN but there is no EN–ES MT available, they could use the EN–IT system and their linguistic intercomprehension abilities.

3 Methodology

We explore the four research questions above using closure or *cloze* test methodology, a method that involves filling in gaps corresponding to single words that have been removed from an extract from a written text.[12] Our study is grounded on these three pillars: (a) reading-comprehension questionnaires with questions like *Who was the president of the Green Party in 2011?* have repeatedly been used to evaluate the usefulness of machine-translated text (Jones et al., 2005, 2009, 2007; Berka et al., 2011; Weiss and Ahrenberg, 2012); (b) cloze testing or gap-filling has extensively been used as an alternative way to measure reading comprehension (Rankin, 1959; Page, 1977); and (c) gap-filling may sometimes be considered to be roughly equivalent to question answering: *In 2011, _____ was the president of the Green Party.* Therefore, we work upon the assumption that gap-filling success measures the usefulness of machine-translated text for comprehension. Inspired in previous work (O'Regan and Forcada, 2013; Trosterud and Unhammer, 2012; Ageeva et al., 2015), the method was used for 65- to 75-word-long target texts previously translated by professionals with gaps every fifth word.[13]

Subjects (native ES speakers with an EN, FR and IT level equal to or lower than CEFRL B1 were provided with different kinds of hints to help them fill in the gaps in professionally translated ES text. The aim was to determine whether the hint was useful for comprehension, as measured by the rate of success in filling out the gaps (that is, the fraction of correctly filled gaps) in the incomplete professionally-translated ES text.

Forty-four test texts were taken from 4 different sources with different degrees of specialization, and were tested in four different hinting situations, distributed as follows:

a) *Using machine translation into L_1 as a hint*: EN–ES and FR–ES MT output[14] — one highly specialized text on natural sciences ('NAT'), one highly specialized text on human and social sciences ('SOC'), one journalistic or informative semi-specialized text ('INF'), and one non-specialized or general text ('NO') that had been translated by the three MT systems, both from EN to ES and from FR to ES (total, 24 texts);

b) *Using text in $L' \simeq L_1$ as a hint*: native or professionally-translated IT texts — four texts written by an IT native professional translator from four EN sources with the same degree of specialization;

[12] Gaps should be filled with either the exact word removed from the source text, or with a synonym or a functionally equivalent unit, that is, a lexical unit that creates a target text with the same meaning as the source in that specific context.

[13] Preliminary experiments showed that the readers' gap-filling performance showed a marked reduction when gaps occurred every five words. Note also that (O'Regan and Forcada, 2013) poke holes with a probability, not periodically as it is done here, but this may be considered be equivalent.

[14] These translations were retrieved in November–December 2014 and it is worth mentioning that online MT systems change over time as models get re-freshed or even when a technological change occurs (such as the recent switch from statistical MT to neural MT for EN–ES and EN–IT)

EN source text (not shown to subjects)	If no formal authorisation has been given by the host state, a third-country national's presence may be considered unlawful by that state. Both EU and ECHR law, however, set out circumstances in which a third-country national's presence must be considered lawful, even if unauthorised by the state concerned.
Hint: Machine-translated ES text	Si no se ha dado ninguna autorización formal por el estado de anfitrión, la presencia de un nacional de terceros países se puede considerar ilegal por ese estado. La ley de la UE y del ECHR, sin embargo, estableció las circunstancias en las cuales la presencia de un nacional de terceros países se debe considerar legal, incluso si es desautorizado por el estado trató.
Problem: Professionally-translated ES text with gaps	Si el Estado de acogida no le ha concedido una autorización formal, dicho Estado **[...]** considerar que la presencia **[...]** un nacional de un **[...]** país es irregular. Sin **[...]**, tanto el Derecho de **[...]** UE como el CEDH **[...]** circunstancias en las que **[...]** presencia de un nacional **[...]** un tercer país se **[...]** considerar legal, aunque el **[...]** miembro de que se trate no la haya autorizado.

Figure 1 – An example gap-filling problem: the subject has to fill with a single word the gaps (one every 5 words) introduced in the professionally-translated text, using the machine-translated text as a hint. The source text is not shown. The solutions in this case are *puede*, *de*, *tercer*, *embargo*, *la*, *establecen*, *la*, *de*, *debe*, *Estado*.

c) *Using machine translation into $L' \simeq L_1$ as a hint*: EN–IT MT output — eight texts from the same four sources that were translated from EN into IT by two of the MT systems (Google and Systran, as Apertium does not provide this combination);

d) *Using non-natively produced L_1 text as a hint*: either an EN text translated into ES by an EN native speaker or a FR text translated into ES by a FR native speaker (both with a ES B2 level according to the CEFRL) — four texts from the same four sources translated from EN into ES by a native speaker of EN and four texts from the same four sources translated from FR into ES by a native speaker of FR (8 texts in total).

Figure 1 shows an example gap-filling problem in which a machine translated text from EN is shown as a hint, that is, an example of the first hinting situation.

All the texts were extracted from institutional publications, that is, works published or translated by staff for linguists in international institutions or organizations with the exception of the non-specialized or general texts, which came from different translations of the Bible in the languages involved. "NAT" texts were taken from the different language versions of WHO document http://www.who.int/pehemf/publications/en/EMF_Risk_ALL.pdf. "SOC" texts were taken from the FRA–EHCR–Council of Europe *Handbook on European law relating to asylum, borders and immigration* (http://fra.europa.eu/sites/default/files/handbook-law-asylum-migration-borders-2nded_en.pdf). "INF" texts were taken from the European Commission publication (http://ec.europa.eu/economy_finance/publications/general/pdf/the_road_to_euro_poster_en.pdf). "NO" texts were taken from four editions of the Bible (books of Job and Esther) in different languages —the Spanish *Dios habla hoy* (1996), the English *Easy-to-Read Version* (1987), the French *Bible Segond 21* (2007) and the Italian *Bibbia Diodati* (1991).

The test was taken by 71 native Spanish undergraduate students from either the first year of the Degrees in Journalism, Audiovisual Communication, Advertising and Public Relations, Translation and Intercultural Communication, Infant Education and Primary Education, or final-

year secondary students. All had only elementary knowledge or, in some cases, no knowledge at all, of EN, FR and IT (that is, a level equal to or lower than CEFRL B1). Students were asked to read the hint and, for the whole test, write a single word in each gap in all the texts. Each job contained 10 gaps; each student completed 22 jobs on average. The fraction of gaps that the students were able to fill successfully (either with the exact word removed or with a synonym) were considered as a measure of usefulness of the text used as a hint (in a scale from 0 to 1) and compared.

4 Results

The results of the test were compared in three blocks: 1) Systran and Google EN–IT and Systran and Google FR–ES MT output with IT texts written by a native speaker for reference; 2) Systran, Google and Apertium MT EN–ES output with texts written in ES by a native EN speaker for reference; 3) Systran, Google and Apertium FR–ES MT output with texts written in ES by a native FR speaker for reference. In each block, five comparisons were made: overall or general (that is, without considering the text type or its degree of specialization), NAT, SOC, INF and NO (see section 3).

In Tables 1 to 5, language combinations for each MT system or the other texts used as a hint were ordered according to the level of usefulness based on the results of the test undertaken by the students. Different data are given: ('m'), the average fraction of gaps correctly filled by the subjects on a scale of 0 to 1; its variance ('var'), the number of observations in each case ('n'), and the estimated probability that the established order for each group was due to chance and not to a real superiority. In other words, the arithmetic mean in each case is used to establish a ranking of hints as regards their usefulness for comprehension; the probability that the hypothesis behind that order or preference is false is also indicated.[15] As usual, when the latter probability shown on each table between two text squares is lower than 5%, we will interpret that the level of usefulness of the first text with respect to the second is clearly higher and not the result of luck.

4.1 Comparing MT output to text in a closely related language

The results in Table 1 answer the first two research questions, that is: RQ1) whether, in terms of usefulness, MT output and a text written by a native speaker in a language L' in the same language family as the reader's first language L_1 are similar; and RQ2) whether the MT output in a language L' in the same language family as the reader's first language L_1 could be useful for comprehension when translating texts originally written in a language from a different language family and much less connected to their first language.

According to these results, in general terms, a human-written IT text is less useful for comprehension than the MT output resulting from translating a text from a related language (that is, FR into ES; and it is basically as useful for comprehension as the MT output resulting from translating a text from a language belonging to a different language family (that is, EN) into IT.

Strangely, however, for NAT texts, a human-written IT text would be marginally more useful for comprehension to a native ES reader than Google FR–ES MT output; and Google EN–IT MT output would also seem marginally more useful for comprehension than Systran FR–ES MT output and clearly more useful than Google FR–ES MT output. Furthermore, for SOC texts, a human-written IT text is as useful for comprehension as Google FR–ES MT output but much more useful than Systran FR–ES MT output. AS to INF and NO texts, human-written IT texts are much less useful than any of the FR-ES MT outputs.

As stated by Jordan-Núñez (2015), the fact that usefulness of NAT IT texts is very similar to that of MT system output, but considerably higher for SOC texts could be due, perhaps, to the

[15] The probability of the null hypothesis (both averages being equal) has been computed using Welch's two-tail *t*-test, https://en.wikipedia.org/wiki/Welch's_t-test.

TOTAL	NATURAL SCIENCES ('NAT')	SOCIAL SCIENCES ('SOC')	JOURNALISTIC ('INF')	NOT SPECIALIZED ('NO')
FR→ES GOOGLE m = 0.7218 var = 0.0254 n = 142	**EN→IT GOOGLE** m = 0.8467 var = 0.0079 n = 30	**ITALIAN** m = 0.6667 var = 0.0354 n = 30	**FR→ES SYSTRAN** m = 0.8463 var = 0.0120 n = 41	**FR→ES GOOGLE** m = 0.8098 var = 0.0239 n = 41
74,30% ↓	10.38% ↓	54.39% ↓	11.86% ↓	41.18% ↓
FR→ES SYSTRAN m = 0.7141 var = 0.0537 n = 142	**FR→ES SYSTRAN** m = 0.7667 var = 0.0451 n = 30	**FR→ES GOOGLE** m = 0.6400 var = 0.0218 n = 30	**EN→IT SYSTRAN** m = 0.7951 var = 0.0310 n = 41	**FR→ES SYSTRAN** m = 0.7829 var = 0.0195 n = 41
4.43% ↓	21.91% ↓	40.23% ↓	17.32% ↓	**0.03%** ↓
EN→IT GOOGLE m = 0.6634 var = 0.0357 n = 142	**ITALIAN** m = 0.7067 var = 0.0248 n = 30	**EN→IT SYSTRAN** m = 0.6033 var = 0.0348 n = 30	**FR→ES GOOGLE** m = 0.7488 var = 0.0156 n = 41	**EN→IT GOOGLE** m = 0.6463 var = 0.0330 n = 41
3,39% ↓	13.64% ↓	15.05% ↓	**1.87%** ↓	**0.13%** ↓
ITALIAN m = 0.6120 var = 0.0469 n = 142	**FR→ES GOOGLE** m = 0.6467 var = 0.0226 n = 30	**EN→IT GOOGLE** m = 0.5400 var = 0.0318 n = 30	**ITALIAN** m = 0.6805 var = 0.0176 n = 41	**EN→IT SYSTRAN** m = 0.5146 var = 0.0308 n = 41
49.10% ↓	**0.00%** ↓	**0.04%** ↓	31.74% ↓	8.76% ↓
EN→IT SYSTRAN m = 0.5937 var = 0.0532 n = 142	**EN→IT SYSTRAN** m = 0.4167 var = 0.0401 n = 30	**FR→ES SYSTRAN** m = 0.3867 var = 0.0274 n = 30	**EN→IT GOOGLE** m = 0.6439 var = 0.0365 n = 41	**ITALIAN** m = 0.4341 var = 0.0578 n = 41

Table 1 – Gap-filling success rates for MT output compared with those for an Italian text written by a native Italian speaker (shaded boxes) and with those for MT output resulting from translating from English into Italian.

deficiencies in the glossaries in the MT systems given the appreciable conceptual and lexical variety within these fields (especially in law, due to partial or even false equivalence in legal terms or institution names).

Likewise, in general terms, the FR–ES MT output is more useful for comprehension than EN–IT MT output (see RQ2). However, depending on the degree and the field of specialization of the text and the MT system used, EN–IT MT output could be more useful to a native Spanish reader.

More specifically, for NAT texts, Google EN–IT MT output would appear to be more useful than the FR–ES MT output, and Systran EN–IT MT output is less useful. For SOC texts, any of the EN–IT outputs is slightly less or as useful than Google FR–ES MT output but far more useful than FR–ES Systran output.

For INF texts, Systran EN–IT MT output would be slightly less useful for comprehension than Systran FR–ES[16] but would be slightly more useful for comprehension than the output resulting from translating a text in the same language combination with Google. The output from translating a text from English into Italian with Google is less useful for comprehension than the output from the same MT system when translating from French into Spanish.

Lastly, for NO texts, all of the EN–IT MT outputs are less useful for comprehension than any of the FR–ES outputs.

[16]Note, however, that because two MT systems are made by the same company they do not have to be similar at all. For instance, while Apertium systems are structurally very similar to each other, their performance varies widely with the language pair or the level of development.

4.2 Comparing MT output with non-native writing in the target language

The results shown in Tables 2 and 3 aim at answering the third research question (RQ3), that is, to demonstrate whether MT output and a text written in the reader's first language L_1 by non-native advanced 'independent speakers' —with a CEFRL B2 EN or FR level— are similar to a native ES reader as regards intelligibility.

TOTAL	NATURAL SCIENCES ('NAT')	SOCIAL SCIENCES ('SOC')	JOURNALISTIC ('INF')	NOT SPECIALIZED ('NO')
EN→ES APERTIUM m = 0.7718 var = 0.0354 n = 142	**EN→ES APERTIUM** m = 0.7933 var = 0.0331 n = 30	**EN→ES GOOGLE** m = 0.6533 var = 0.0502 n = 30	**EN→ES APERTIUM** m = 0.8707 var = 0.0206 n = 41	**EN→ES APERTIUM** m = 0.8268 var = 0.0165 n = 41
2.13% ↓	62.32% ↓	95.13% ↓	34.70% ↓	**3.22%** ↓
EN→ES GOOGLE m = 0.7218 var = 0.0308 n = 142	**EN→ES GOOGLE** m = 0.7700 var = 0.0339 n = 30	**EN→ES SYSTRAN** m = 0.6500 var = 0.0384 n = 30	**EN→ES SYSTRAN** m = 0.8463 var = 0.0065 n = 41	**EN→ES GOOGLE** m = 0.7537 var = 0.0195 n = 41
0.19% ↓	19.64% ↓	**1.11%** ↓	**0.00%** ↓	**0.00%** ↓
EN→ES SYSTRAN m = 0.6493 var = 0.0454 n = 142	**ES ANGLO** m = 0.7167 var = 0.0159 n = 30	**EN→ES APERTIUM** m = 0.5400 var = 0.0135 n = 30	**EN→ES GOOGLE** m = 0.7049 var = 0.0115 n = 41	**ES ANGLO** m = 0.5537 var = 0.0385 n = 41
1.24% ↓	6.34% ↓	28.04% ↓	**0.02%** ↓	**2.51%** ↓
ES ANGLO m = 0.5930 var = 0.0258 n = 142	**EN→ES SYSTRAN** m = 0.6367 var = 0.0279 n = 30	**ES ANGLO** m = 0.5100 var = 0.0092 n = 30	**ES ANGLO** m = 0.6024 var = 0.0157 n = 41	**EN→ES SYSTRAN** m = 0.4610 var = 0.0289 n = 41

Table 2 – Gap-filling success rates for MT output compared with those for a Spanish text written by a native English speaker ("ES ANGLO", shaded boxes).

TOTAL	NATURAL SCIENCES ('NAT')	SOCIAL SCIENCES ('SOC')	JOURNALISTIC ('INF')	NOT SPECIALIZED ('NO')
ES FRANCO m = 0.7789 var = 0.0231 n = 142	**ES FRANCO** m = 0.8267 var = 0.0186 n = 30	**ES FRANCO** m = 0.6400 var = 0.0225 n = 30	**FR→ES SYSTRAN** m = 0.8463 var = 0.0120 n = 41	**ES FRANCO** m = 0.8293 var = 0.0246 n = 41
0.00% ↓	9.63% ↓	100.00% ↓	**2.31%** ↓	57.22% ↓
FR→ES GOOGLE m = 0.7218 var = 0.0254 n = 142	**FR→ES SYSTRAN** m = 0.7667 var = 0.0451 n = 30	**FR→ES GOOGLE** m = 0.6400 var = 0.0218 n = 30	**ES FRANCO** m = 0.7951 var = 0.0080 n = 41	**FR→ES GOOGLE** m = 0.8098 var = 0.0239 n = 41
74.30% ↓	**1.46%** ↓	**2.06%** ↓	5.70% ↓	41.18% ↓
FR→ES SYSTRAN m = 0.7141 var = 0.0537 n = 142	**FR→ES GOOGLE** m = 0.6467 var = 0.0226 n = 30	**FR→ES APERTIUM** m = 0.5333 var = 0.0382 n = 30	**FR→ES GOOGLE** m = 0.7488 var = 0.0156 n = 41	**FR→ES SYSTRAN** m = 0.7829 var = 0.0195 n = 41
0.00% ↓	72.94% ↓	**0.27%** ↓	**0.00%** ↓	**0.00%** ↓
FR→ES APERTIUM m = 0.5268 var = 0.0499 n = 142	**FR→ES APERTIUM** m = 0.6300 var = 0.0463 n = 30	**FR→ES SYSTRAN** m = 0.3867 var = 0.0274 n = 30	**FR→ES APERTIUM** m = 0.5854 var = 0.0218 n = 41	**FR→ES APERTIUM** m = 0.3878 var = 0.0616 n = 41

Table 3 – Gap-filling success rates for MT output compared with those for a Spanish text written by a native French speaker ("ES FRANCO", shaded boxes)

English into Spanish: According to these results, in general terms, for a native ES reader, the results show that EN–ES MT output is more useful for comprehension than a text written in ES by a native EN speaker ("ES ANGLO" in Table 2) with a CEFRL B2 level in ES. However, the text written by that native EN speaker would seem to be marginally more useful than Systran MT system output for NAT or NO texts.

French into Spanish: Overall, texts written in ES by a native FR speaker with a CEFRL B2 level in ES are more useful than any FR–ES MT output. However, the text written by the native French speaker ("ES FRANCO" in Table 3) is more useful for comprehension than most FR–ES MT output for NAT texts, as useful as Google FR–ES output for SOC texts, at least almost as useful as any FR–ES MT output for NO texts, and less useful than Systran MT FR–ES output for INF texts, but more useful than Google or Apertium FR–ES output. This may arise from the smaller differences between languages from the same family allowing a non-native speaker to write a text with fewer grammar and lexical mistakes, since the two languages involved share common vocabulary and structures. Alternatively, in any case, non-native mistakes do not prevent a native Spanish reader from understanding the text.

4.3 Comparing MT systems

When comparing data from the three MT systems, it has been shown, as already stated in Jordan-Núñez (2015), that, in general terms, the usefulness of any MT EN–ES output resulting from using the three MT systems is similar. However, the usefulness of the Apertium MT output is slightly larger. This is not the case for FR–ES: the usefulness of Apertium output is considerably lower.

More precisely, for highly specialized NAT texts (see table 4), Apertium seems to be the best EN–ES MT system for NAT texts, on par with Google, while Systran seems to be the best FR–ES system. Likewise, Google EN–ES and FR–ES are apparently the best MT systems for SOC texts.

For semi-specialized, INF texts (see Table 5), Apertium seems to be the best EN–ES MT system, on par with Systran, while Systran is the best FR–ES MT system. Equally, for NO texts (see Table 5), Apertium is again the best MT EN–ES system, and Google seems to be the best MT FR–ES system, on par with Systran.

5 Discussion

In view of the cloze-test results reported, one can answer the research questions posed and prove the hypotheses described:

RQ1) As already indicated in Jordan-Núñez (2015), to a native ES reader, the usefulness of a text written in a language L' from the same language family as the reader's L_1 (in this case, $L' =$IT) is higher for highly specialized texts than for general or non-specialized texts, as a consequence of the higher density of international vocabulary in NAT, and to a lesser extent, SOC texts. However, although the usefulness of a human-written IT text is very similar to MT output for highly specialized NAT texts, the IT text would appear to be more useful than any of the MT outputs studied for SOC texts. This may be the result, as mentioned above, of the deficiencies of the glossaries in the MT systems given the appreciable conceptual and lexical variety within these fields.

RQ2) Although, generally speaking, any of the FR–ES MT outputs studied is more useful for comprehension to a native ES speaker than EN–IT MT output, usefulness depends on the level and the field of specialty, and on the MT system. In many cases, especially when translating highly specialized material, EN–IT MT output may be more useful than FR–ES MT output– for example, when using Google to translate NAT texts. This leads to the conclusion that MT output in a language L' in the same family as the reader's L_1 –e.g., IT for a native ES speaker– could be used to facilitate their comprehension of texts originally written in a language from a different family and, evidently, far removed from their first language (that is, EN). Some findings, however, as already stated, do not apply to MT in general, but to limitations in the MT system that lead to differences in performance across text genres.

RQ3) As stated above, in general terms and to a native Spanish reader, EN–ES MT output is

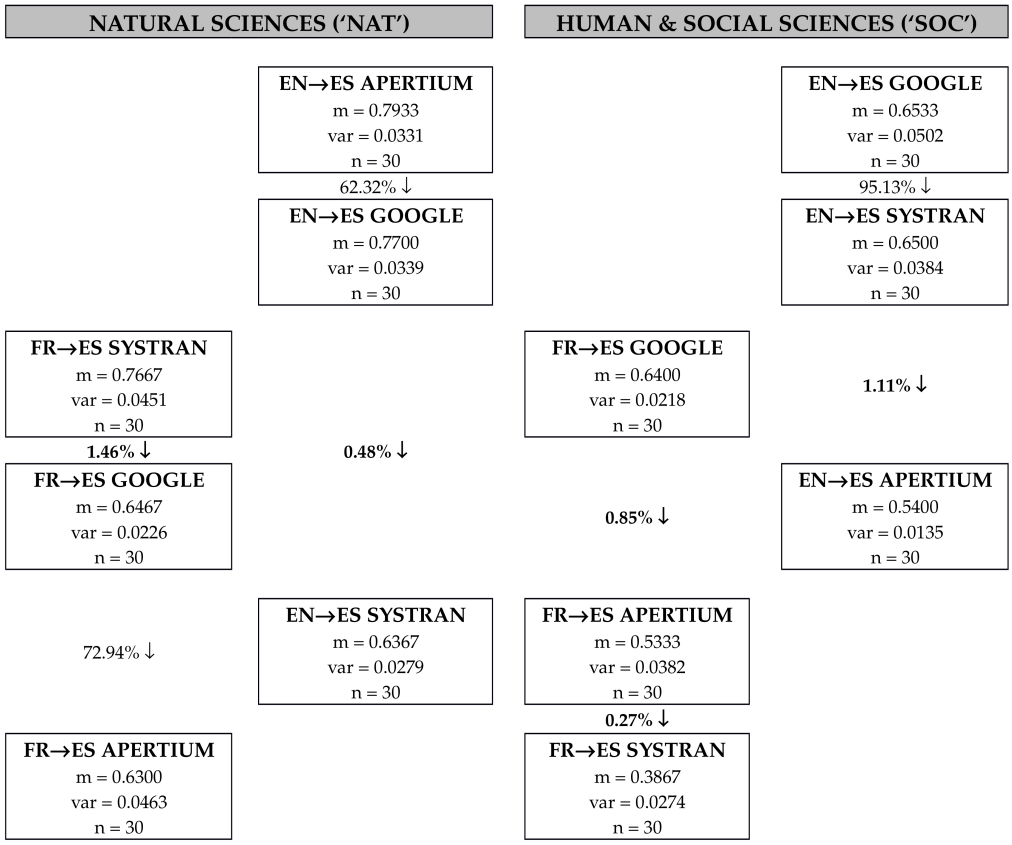


Table 4 – Comparing the gap-filling success rate for the output of different MT systems, for highly specialized texts.

more useful for comprehension than a text written in ES by a native EN speaker. However, texts written in ES by a native FR speaker are more useful than any of the FR–ES MT outputs. In fact, the text written in ES by the native FR speaker is much more useful than the text written by the native EN speaker (considering that both speakers have the same level of ES; that is, CEFRL B2). This could be due to that there is less interference or carry-over from FR when writing in ES (especially in syntax) given that both languages belong to the same language family.

RQ4) Although, in general terms, Apertium seems to be better for EN–ES than the other two MT systems and considerably worse for FR–ES, this also depends, to a greater or lesser extent, on the degree and field of specialty, and on the MT system used. In these experiments, Apertium is the best EN–ES MT system for any text type except SOC, which are best translated by Google and Systran. Google is the best FR–ES MT system for SOC and NO texts, while Systran is the best FR–ES MT system when dealing for NAT and INF texts.

5.1 Critical appraisal of the methodology

It is important to formulate critical comments regarding the methodology used in this pilot study, to be taken into account when pursuing further research.

It has been noticed that the fraction of gaps corresponding to function, structure or "stop" words — that is, articles, pronouns, prepositions or conjunctions— varies from one text type to

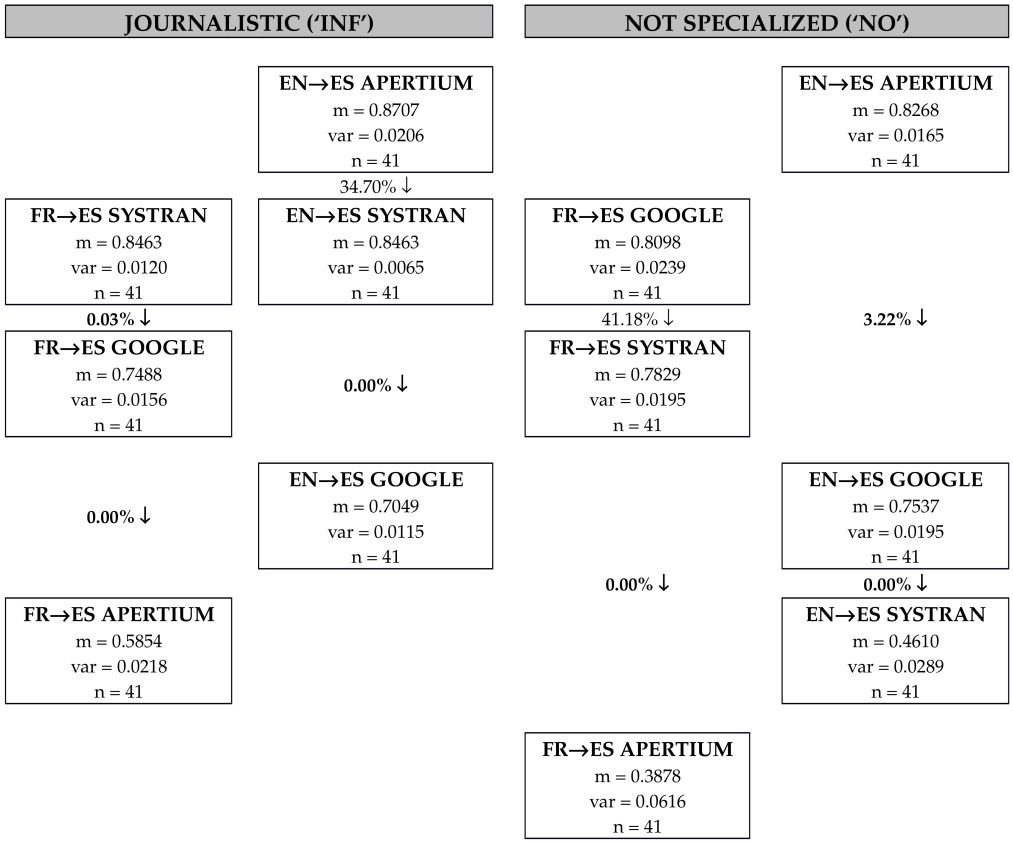


Table 5 – Comparing the gap-filling success rate for the output of different MT systems, for informative or journalistic texts and non-specialized texts

another. This could have a considerable effect on the results. In order to avoid this, in the test for the final project, the gaps should be done just on content words, that is, avoiding gaps at "stop" words (as was done by O'Regan and Forcada (2013)).

The results may also have been affected by the source of texts chosen when designing the test. It has been noticed that the Spanish version of the NAT text used a Latin-American variety of Spanish, which may have been less recognizable to a group of students familiar with Castilian Spanish; this should have been avoided when designing the research plan, as language varieties introduce an uncontrolled variable within the experimental design. Likewise, the Bible may have not been a good choice, since the different versions used differ quite noticeably from each other and, actually, they cannot strictly be considered mutual translations.[17] In order to avoid this, in the final project, all the texts should be taken from publications by institutional organizations using Castilian Spanish and where writing/editing and/or translation processes are followed by proofreading or quality assurance processes. In the case of non-specialized texts, they could also be taken from lesser-known novels that have been translated into all the languages involved in the study, to avoid the risk that the student recognizes the text and effortlessly fills the gaps without using information from the hint (as found by O'Regan and Forcada (2013) when no hint

[17] In fact, some passages have been translated so differently in each language version that the students could have found it difficult to find the information to help them fill the gaps from the hint given.

was given).

6 Conclusions and future work

As previously indicated, the broader research within which this study is framed seeks to identify whether the differences between a professional translation and MT output are similar to the differences between languages within a language family and, then, explore whether intercomprehension strategies (Klein and Stegmann, 2000; Martín Peris et al., 2005) are useful to avoid those non-human traits and understand the main message of the text. However, it is fair to say that the subjects participating in this study could have made use of spontaneous intercomprehension abilities not necessarily corresponding to the strategies described by the above authors and that some of those strategies may not apply to machine-translated texts.

As shown above, it should be admitted that the usefulness of a text written in a language from the same language family as the reader's L_1 is higher for highly specialized than for non-specialized texts. However, regarding MT output, this level of usefulness depends quite often on the level of specialty but also on the field and on the MT system used.

In view of the results, it seems also reasonable to postulate that MT output into a language L' in the same family as the reader's first language L_1 could be used to facilitate their comprehension of texts originally written in a language from a different family. Likewise, EN–ES MT output is more useful for comprehension than a text written in ES by a native EN speaker, while texts written in ES by a native FR speaker are more useful than FR–ES MT output.

Finally, in the future, it will be interesting to assess to what extent the skills used to understand MT output and the linguistic intercomprehension skills used by the methods designed are similar. If EN–ES MT output is, in general terms, as useful for comprehension as an IT text (depending, however, on the MT system used, the language combination and the level of specialization) and that MT output has a common vocabulary and some common morpho-syntactic structures (despite containing some mistakes that distance it from a native human-written text), one could argue that (generally unconscious) linguistic skills used in linguistic intercomprehension may be similar to those used to understand MT output. To shed some light on this, we will classify, label and quantify the types of machine translation errors or disfluencies and study their effect in the comprehension process, distinguishing errors or disfluencies that may be taken to be similar to the divergences observed between languages in the same family from those that are not. For instance, when a source word is out of the vocabulary of the MT system and left untranslated, but it is however similar to the target word, its negative impact should be less severe than in the case in which the word is very different.

Acknowledgements: KJN thanks Prof. Joan Fonollosa for his help in determining the statistical significance, and San Jorge University and Pompeu Fabra University for allowing him to have a research stay in the latter in order to work on this project. MLF thanks the Universitat d'Alacant for support during a sabbatical stay at the Universities of Sheffield and Edinburgh and acknowledges the support of the Spanish Ministry of Economy and Competitiveness through grant TIN2015-69632-R and the Spanish Ministry of Education, Culture and Sport through grant PRX16/00043. EC thanks the Spanish Ministry of Economy and Competitiveness for support through grant FFI2016-76245-C3-3-P. We also thank Dr. César Rodríguez, Dr. Naswha Nashaat, Dr. Manuel Gómez, and lecturers Santiago Lamas, María Pilar Cardos and Rachel Harris from San Jorge University and Dr. Elena Alcalde from the University of Granada for allowing us to test their students.

References

Ageeva, E., Tyers, F. M., Forcada, M. L., and Pérez-Ortiz, J. A. (2015). Evaluating machine translation for assimilation via a gap-filling task. In *Proceedings of EAMT*, pages 137–144.

Berka, J., Černý, M., and Bojar, O. (2011). Quiz-based evaluation of machine translation. *The Prague Bulletin of Mathematical Linguistics*, 95:77–86.

Clua, E. (2003). Eurocom: ciutadans plurilingües per a una europa multilingüe. *Enxarxa't: revista de la Xarxa de Dinamització Lingüística de la UB*, 2:4–5.

Jones, D., Gibson, E., Shen, W., Granoien, N., Herzog, M., Reynolds, D., and Weinstein, C. (2005). Measuring human readability of machine generated text: three case studies in speech recognition and machine translation. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v:1009–v:1012. IEEE.

Jones, D., Herzog, M., Ibrahim, H., Jairam, A., Shen, W., Gibson, E., and Emonts, M. (2007). Ilr-based mt comprehension test with multi-level questions. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 77–80. Association for Computational Linguistics.

Jones, D., Shen, W., and Herzog, M. (2009). Machine translation for government applications. *Lincoln Laboratory Journal*, 18(1).

Jordan-Núñez, K. (2015). ¿es igual de comprensible la salida de un sistema de ta que un texto escrito en otra lengua de la misma familia de lenguas? In LEXITRAD, editor, *New Horizons in Translation and Interpreting*. Tradulex.

Klein, H. G. and Stegmann, T. D. (2000). *EuroComRom - Die sieben Siebe: Romanische Sprachen sofort lesen können*. Shaker.

Martín-Peris, E. (2011). La intercomprensión: concepto y procedimientos para su desarrollo en las lenguas románicas. In de Zarobe, Y. R. and de Zarobe, L. R., editors, *La lectura en lengua extranjera*, pages 246–271. Portal Education.

Martín Peris, E., Clua, E., Klein, H., and Stegmann, T. (2005). *EuroComRom — Los siete tamices: Un fácil aprendizaje de la lectura en todas las lenguas románicas*.

Meissner, F. (2004). Esquisse d'une didactique de l'eurocompréhension. In *EuroComRom: Les sept tamis: lire les langues romanes dès le départ; avec une esquisse de la didactique de l´eurocompréhension*, pages 19–83.

O'Regan, J. and Forcada, M. L. (2013). Peeking through the language barrier: the development of a free/open-source gisting system for Basque to English based on apertium.org. *Procesamiento del Lenguaje Natural*, pages 15–22.

Page, W. D. (1977). Comprehending and cloze performance. *Literacy Research and Instruction*, 17(1):17–21.

Rankin, E. F. (1959). The cloze procedure: its validity and utility. In *Eighth yearbook of the National Reading Conference*, volume 8, pages 131–144. Milwaukee: National Reading Conference.

Trosterud, T. and Unhammer, K. B. (2012). Evaluating North Sámi to Norwegian assimilation RBMT. In *Proceedings of the Third International Workshop on Free/Open-Source Rule-Based Machine Translation (FreeRBMT 2012)*.

Weiss, S. and Ahrenberg, L. (2012). Error profiling for evaluation of machine-translated text: a Polish-English case study. In *LREC*, pages 1764–1770.

Machine Translation as an Academic Writing Aid for Medical Practitioners

Carla Parra Escartín carla.parra@adaptcentre.ie
Sharon O'Brien sharon.obrien@dcu.ie
ADAPT Centre, SALIS, Dublin City University, Dublin, Ireland
Marie-Josée Goulet marie-josee.goulet@uqo.ca
University of Quebec in Outaouais, Gatineau, Canada
Michel Simard Michel.Simard@cnrc-nrc.gc.ca
National Research Council of Canada

Abstract

In this paper we explore the utility of Machine Translation as a writing aid and its impact on the quality of the text produced. We focus on medical practitioners who are native speakers of Spanish and who need to publish their scientific work in English as a foreign language. After carrying out a general survey to determine whether Spanish-speaking medical practitioners already use MT as a writing aid, we engaged five participants in an experiment where we asked them to write a paper in Spanish that was subsequently machine translated. They were then asked to post-edit the MT output. We analyse their post-edits and further attempt to evaluate the overall quality of their texts by engaging a professional proofreader. Our results suggest that the texts produced with the help of MT+post-editing still require many edits in order to be considered of acceptable quality. In the conclusion, we identify several avenues worthy of future investigation and that could help achieve better quality.

1. Introduction

In recent times two developments have led to a new type of Machine Translation (MT) deployment, i.e. MT for personal use. Those two developments are: (1) freely available online MT systems and (2) increasing quality of MT output, for some language pairs at least. The 'average' internet user can now take advantage of MT to assist with various tasks such as school homework, translating website content for service and product reviews, and so on. Embedding of MT widgets in all sorts of websites has also contributed to personal MT usage.

One user type that might avail of MT for personal, and professional, purposes is the academic whose first language is not English, but who, in order to widely disseminate his or her work, wishes to publish in English. It is our belief that some who write in English as a Foreign Language (henceforth: EFL writers) are using freely available online MT systems as an aid to the writing process, first writing passages of text in their L1 (or first language) and translating those into English as they produce academic articles.

Despite increasing quality from MT engines, it is still accepted that MT output generally requires post-editing before it is of publishable quality. The focus of our research is on the use of MT as an aid by EFL writers in specialised fields. As this topic appears to have not been researched in any detail, as outlined below, we aim to explore the utility of MT as a writing aid and its impact on the quality of the text produced.

English is the undisputed *lingua franca* of academia (Bennett, 2013, 2014, 2015). This forces those who are not native speakers of English to publish in English in order to disseminate their research and progress their careers. As we have reported elsewhere (O'Brien et al., forthcoming; Goulet et al., forthcoming), this leads to a considerable disadvantage, especially for those who do not master English as a foreign language. The disadvantage touches on the cognitive level (Breuer, 2015), as well as on the career level, if journal acceptance is taken into consideration (Benfield and Feak, 2006), and on the economic level, if cost of additional translators or proofreaders is factored in (Lillis and Curry, 2010). Using MT as a writing aid might ease some of these disadvantages by (1) allowing people first to write in their L1 and then use MT as an aid to produce text in English, thereby tackling some of the cognitive demands of writing in a foreign language and (2) reducing costs by eliminating the need for translators or proofreaders, who often do not possess the specialised domain vocabulary in any case (Willey and Tanimoto, 2015).

Of course, there are several assumptions here that need to be examined. For example, does writing in L1, Machine Translating, and post-editing by the author (which we term 'self-post-editing') reduce the cognitive burden on the EFL writer? Can non-translators (authors in our context) post-edit their own work to an adequate level of quality? Does this method lead to higher quality in the English text, such that journal acceptance is a smoother process? Does it eliminate the need for a proofreader? We cannot tackle all of these questions here, but we have begun to address the questions regarding the quality of the English text (see below), the need for a proofreader, and the feasibility of self-post-editing.

2. Related Research

We report more fully in O'Brien et al. (forthcoming) and Goulet et al. (forthcoming) on related research and so will just summarise here. To put it succinctly, there is little work that focuses on this topic. Some work has been done on MT and second-language writing (for example, Niño, 2008, Garcia and Pena, 2011; and O'Neill, 2012) that demonstrates that MT can be useful as a second-language writing support. This previous work focuses mostly on university students who were learning languages. To the best of our knowledge, no work has been done concerning MT as an aid for professional writing.

In O'Brien et al. (forthcoming), we made a first attempt to explore the utility of MT for the EFL academic cohort. This exploration found that the median times for drafting abstracts were not substantially different between L1 and EFL, however the revision times and number of revisions implemented were greater for the L1 (+MT) sections. The participants were split more or less down the middle in terms of their perceptions of ease of task, while six (out of nine) felt that the quality produced was equal for both and three thought that writing in EFL produced better quality. A professional proofreader was hired to evaluate the quality of the texts, and her assessment supported the authors' perception of quality. In short, we found that there was encouraging support for the assumption that MT could be used as a writing aid by EFL writers without taking up significantly more time and without impacting on quality.

In Goulet et al. (forthcoming), we analysed this data set in more detail, comparing the edits implemented by the proofreader across both halves of the abstract in order to ascertain whether the editing required for text produced in EFL was different from that written in L1 and subsequently machine translated and self-post-edited. In summary, we found the number of edits to be similar (5% and 6% of the total word count in EFL and MT respectively), but that for the authors with Arabic and Chinese as L1, the number of edits to the MT'd parts were higher than for languages such as French or Spanish. Overall, there were no very outstanding differences in terms of the proofreader's edits when one part of the abstract was compared to the other,

again indicating that MT as an academic writing aid certainly does not have a negative impact on the quality of the text produced.

3. Motivation

The exploratory study summarised in the previous Section 2 provided impetus for a follow-up study, which is the focus of this paper. Having previously recruited participants from a broad range of disciplines and languages, it was decided that it would be relevant to focus on one language pair and on one domain for a more in-depth analysis. Knowing anecdotally that medical practitioners seek to, and often have to, publish their research findings, we decided to focus on them as a new cohort. Moreover, we had anecdotal evidence that medical practitioners with Spanish as an L1 sometimes struggle to write in English. Add to this the fact that MT is known to perform relatively well between Spanish and English and so users might be encouraged by its output, we decided to recruit and analyse self-post-editing within this cohort. Our focus of attention this time was to understand more fully the nature of the self-post-editing task as well as MT usage among medical practitioners in general. We consequently asked the following questions:

1) *Are Spanish-speaking medical practitioners using MT as a personal writing support already?*

This question sought to explore whether or not our assumption about personal MT usage was correct.

2) *Without any training in MT or post-editing, what type of edits do medical practitioners make when they write in Spanish and then machine translate into English and self-post-edit?*
   a) *Are essential edits implemented or ignored?* (See the Methodology discussion in Section 4.2.3 below for a definition of 'Essential Edits' and 'Essential Edits not Implemented')
   b) *How much non-essential (or preferential) editing is carried out?*
   c) *Are errors introduced via self-post-editing?*

Our goal here is to understand the natural competence for self-post-editing without any training whatsoever and to move towards developing potential supports for post-editing for such cohorts. By analysing essential and preferential edits as well as errors introduced we aim at establishing the degree of quality achieved in our experimental setup.

3) *How much editing is required by a professional proofreader on top of the post-edited documents and what type of edits are implemented?*

With this question we investigate whether L1+MT+self-post-editing actually requires another round of proofreading or whether the proofreader could be eliminated from this cycle. Again, this taps into a measurement of the quality produced during the self-post-editing setup.

4. Methodology and Experimental Setup

In order to address our initial research questions (cf. Section 2), we combined different research tools: questionnaires, active writing and post-editing, proofreading, and annotation of the edits made under each condition (self-post-editing and professional proofreading). In what follows we describe the methodology and experimental setup.[1]

[1] The research reported here was granted ethical approval by the relevant Research Ethics Committees.

4.1. General Survey

To address the question: "*Are Spanish-speaking medical practitioners using MT as a personal writing support already?"*, we surveyed medical practitioners in Spain. The survey was run between 19 December 2016 and 27 January 2017 and the link to our questionnaire was sent to many organisations, including medical specialised associations, medical unions, universities and research institutes in Spain. We had a total of 50 responses. The questionnaire addressed several questions, including whether or not the respondents already used MT as a writing aid.

This general questionnaire also helped us to identify potential participants for the experiment. At the very end of our questionnaire, we asked the respondents whether they would be willing to participate in experiments using MT and collected their e-mail addresses. Although 31 of the respondents provided us with their e-mail addresses, only five were finally available for the first experimental cycle, carried out between February and the beginning of April 2017.

4.2. Experimental Setup

4.2.1. Participant Profiles

As stated earlier, only five of our questionnaire respondents (3 men and 2 women) were available to engage in the experiment reported here. Four of them are in an early stage of their careers, are between 20 and 30 years old, and are engaged in their residencies. The fifth one is a researcher at a university or research centre and is between 30 and 40 years old. One specialises in Neurosurgery, another in Internal Medicine, two of them are gynecologists and the fifth one works in Immunology. All of them have Spanish as their mother tongue and all of them speak other languages besides English (Catalan, French, German, Italian and/or Portuguese). Table 1 summarises their self-reported level of English using the Common European Framework of Reference for Languages (CEFR), as well as the level of English as established by an online English test on the Cambridge English website.[2] P01 rated his level of English as lower than what the placement test revealed, whereas P03 rated his level higher. The remaining participants had a fairly accurate self-assessment of their English level.

Participant	Self-Assessment (CEFR, writing)	English Level Test
P01	B2	C1-C2
P02	B1	B1
P03	C1	B2
P04	B2	B2-C1
P05	B1	B1-B2

Table 1: Participants' Level of English

Except for P01, all of the other participants had published a paper before, their number of publications ranged from 1 (P03) to 15 (P05), and only two of them (P02 and P05) had

[2] In order to cross-check their self-assessment with their actual English level, participants were asked to complete an English level test of 25 questions and let us know their final results. The test can be found here: http://www.cambridgeenglish.org/test-your-english/general-english/

published papers in English before. While P02 had only published one paper, P05 had published up to 5 papers in English. In both cases, their reported strategy for publishing was the same: they wrote directly in English and subsequently carried out a self-revision. P02 acknowledged having used Google Translate as a writing aid to confirm the translation of individual words or sentences.

4.2.2. Phase 1 : Publication Drafting in Spanish

We asked our participants to send us a publication or a section of a publication of approximately 750 words that they had originally written in Spanish. We additionally asked them to try, whenever possible, to avoid writing sentences longer than 20 words as this should help to achieve better quality from the MT system. As we could not expect them to count the words in each single sentence, we gave them a visual indication of 20 words in Spanish as being more or less equal to 1.5 lines in MS Word (Times New Roman, font size 12). We aimed at analysing the discussion section, or the section most similar to that, as it is more discursive than other sections (Skelton and Edwards, 2000). We deemed that this section may be one of the most challenging to write, especially for non-native speakers, and that therefore it is a good section to use in testing the use of MT as a writing aid.

4.2.3. Phase 2 : MT and Self-Post-Editing

Upon reception of the texts, we used Google Translate to translate them into English and sent them back to their respective authors asking them to correct the MT output. If they had sent the whole paper, we returned the whole paper translated, and asked them to review the specific section we had selected for our study. We asked them to carry out the revision using the "track changes" functionality in MS Word with the aim of being able to study the types of edits they had made.

After they had returned their self-post-edited texts, we asked them to fill in a post-task questionnaire about their experience. The results of this questionnaire are summarised, together with our analysis, in Section 5.2.

Upon reception of all files, we sought to answer our second research question: "*Without any training in MT or post-editing, what type of edits do medical practitioners make when they write in Spanish and then MT into English and self-post-edit?*". To do so, we annotated all edits made by the medical practitioners. One author annotated the files and highlighted any unclear cases, and subsequently another author went through all the annotations and we carried out a negotiation phase to determine the final annotations in each case. Unclear cases were further discussed with a third author. As at this stage we were mainly interested in determining whether or not medical practitioners were implementing essential or preferential edits and whether new errors were introduced in the self-post-editing process, we chose to adopt the typology proposed by de Almeida (2013), who was interested in the nature of post-edits implemented by professional translators in an attempt to describe what a 'good' post-editor did. De Almeida reviewed many typologies for the analysis of post-editing activity and concluded that there was no internationally adopted model for classifying this type of task. She customised the LISA (2004) and GALE (NIST, 2007) typologies for her own purposes and then layered a number of 'master categories' over this typology. The master categories entail:

- **Essential edits**: if the edit is not implemented, the sentence (or part of it) is either:
  a) Grammatically incorrect (i.e. it obviously breaches a grammatical rule specified in accepted grammar books), or

b) Grammatically correct, but not accurate in comparison to the source text (i.e. it does not contain all the information that is present in the source text, i.e. an omission, or it contains extra information that is not present in the source text, i.e. an unnecessary addition).

- **Preferential edits**: an edit is considered preferential if the sentence from the raw MT output would still be grammatically correct, intelligible and accurate in relation to the source text, even if the edit in question was not implemented.
- **Essential edits not implemented**: This is an essential edit (as defined above) that was not implemented by the author.
- **Introduced errors**: The error was not present in the raw MT output, and it was introduced by the post-editor while editing a sentence. Because of this edit, the sentence (or part of it) is grammatically incorrect and/or inaccurate.

For this paper, we decided to slightly modify these master categories, and thus instead of categorising edits as 'introduced errors', we deemed it important to distinguish between 'introduced errors' that were attempting to correct something (i.e. an edit was essential, but the medical practitioner failed at fixing the problem), or those in which the edit was preferential and resulted in an error. That is: we treated the master category "introduced error" as a subcategory of either "essential edits" or "preferential edits". A more detailed annotation of the types of edits is foreseen for the future.

4.2.4. Phase 3 : Professional Proofreading

As a last stage of our experiment, we recruited a professional translator and proofreader specialised in the medical domain to proofread the texts, after the medical practitioners had post-edited them. We confirmed all the changes made using the "track changes" functionality, and subsequently sent the proofreader the post-edited texts for revision (i.e. we did not provide her with the original Spanish text, nor did we explain the origin of those English texts). As we wanted to avoid over-editing, we also provided her with the following instructions:

> "The texts are written in English and we are looking for a surface revision, that is, pay attention to grammar, orthography, punctuation, syntax, and major stylistic problems. We would like the texts to read well enough to be submitted to a scientific conference, for example.
>
> The texts belong to the medical domain, and are all parts of scientific papers written by doctors."

In order to be able to analyse the proofreader's edits, we requested that the "track changes" functionality in MS Word be used. We then proceeded to annotate these edits using the same typology that we had used to annotate the edits made by the medical practitioners. Although it is true that the typology was meant to be used for the annotation of post-edited texts, we deemed that a classification of essential and preferential edits was also applicable to a proofread text. Thus, we removed the translation dimension from the typology and focused only on the correctness of the text, using the same categories. This strategy allowed us to reply to our last research question: "*How much editing is required by a professional proofreader on top of the post-edited documents and what type of edits are implemented*?".

5. Data Analysis

5.1. General Questionnaire Response

As explained in Section 4.1, we conducted a general survey (in Spanish) seeking to gather information as to how Spanish medical practitioners currently write their publications.

The gender spread was 28 female, 21 male and 1 undeclared. 18 of the respondents were between 20 and 30, 8 between 30 and 40, 4 between 40 and 50, 14 between 50 and 60, and 6 were older than 60.

Most of the respondents were at the beginning of their careers and work in public hospitals. Although there were replies for most of the medical specialties, 20% of replies were from gynaecologists, 16% from cardiologists and another 16% from neurologists.

94% of the respondents indicated that their mother tongue is Spanish. For those who indicated a different mother tongue, all of them stated "Catalan". 84% indicated that they speak English and the remaining 16% indicated that they did not. For self-assessment of English writing skills using the CEFR, only one indicated a C2 level, 4 a C1, 14 indicated a B2 and another 14 B1, 8 chose A2 and 1 A1.

74% of the respondents indicated that they have published scientific papers before and 26% indicated that they have never published.

76% (28 respondents) indicated that they had published papers in English and 24% (nine respondents) said they had no publications in English. The 28 respondents that indicated they had published in English were subsequently asked how those publications were drafted.[3] Nine respondents (32%) indicated that they write directly in English and subsequently ask a colleague or friend with a better level of English to do the corrections. 29% (eight people) indicated that they directly write in English and self-revise, 25% (seven people) indicated that they write in Spanish and hire a professional translator, and another 25% (seven people) indicated that they write directly in English and subsequently hire a proofreader. Two people (7%) said that they hire a professional proofreader if they could do so, and another two acknowledged asking a colleague or friend who is a native speaker of English to do the proofreading. These figures support the claim that some EFL writers feel that they need to seek support from others in order to publish in English. This support is sought from colleagues and/or paid for through professional services.

Of the 28 respondents that had published in English, 19 (68%) indicated they had used Machine Translation for writing academic papers and nine (32%) said they had not. Those who said that they did not use MT (8 respondents, 89%) indicated as the main reason that they did not trust the quality.[4] Two (22%) said that they did not know of the existence of MT, another two indicated that they have problems with terminology, and one indicated "other" and explained that for the type of texts they wrote they were confident enough in English and did not feel the need to use MT.

17 people (89%) indicated they use MT services to check how something is expressed in English, while five (26%) said they used it after drafting a document in Spanish to obtain a preliminary English version they could subsequently post-edit.

Though our questionnaire had a limited number of responses (50), it allowed us to confirm that some Spanish-speaking medical practitioners feel the need to rely on additional supports to aid them in producing articles in English, that some of them are using MT as a writing aid already and that they rarely use it to translate full documents, but rather short passages of text or individual words.

5.2. Post-Task Questionnaire Response

All five participants in our experiment were also asked to fill in a short post-task questionnaire aiming at gathering information about their experience.

[3] This question allowed respondents to select all options that applied to them.
[4] This question allowed respondents to select all options that applied to them.

We first asked them which method for writing scientific publications in English they deemed easiest after their experience participating in our experiment. 60% of them (three participants), chose the option they had just experienced, i.e. writing their publication in their mother tongue and subsequently post-editing an MT version of it. One participant indicated that s/he found it equally easy to write directly in English or to write publications using the proposed workflow, and the fifth participant indicated that s/he preferred to write his/her publications directly in English.

When asked to comment on the difficulties experienced when correcting the MT output, one participant said that s/he had encountered problems with synonyms, and another that s/he had found the translations to be too literal. The third participant said that the MT output was good, the fourth stated that the MT output was better than his/her own English level and therefore s/he found it difficult to identify errors, and the last one said that s/he had encountered the expected issues: grammar problems, terminology and words that change their meaning depending on their context and that had been translated wrongly.

Despite their complaints and comments about the MT output, three out of the five participants deemed that the overall quality of the MT output was at 3 on a scale from 1 to 4, and two gave it the maximum points.

When asked to rank how likely they were to use the proposed workflow for writing scientific papers in the future on a scale from 1 to 4, four of the five participants replied "3", while the fifth replied "2". Three of them further indicated that they thought a second experience like this one would allow them to achieve a better overall quality, whereas two indicated "maybe". Four of them also stated that, with practice, this type of task would become easier, while one said "maybe".

5.3. Word Count Statistics

Word Count Statistics						
Participant	**P01**	**P02**	**P03**	**P04**	**P05**	**TOTAL**
Number of words in ES	1413	759	1058	908	686	4824
Number of words MT (EN)	1340	685	959	865	639	4488
Number of words MT+Self-PE (EN)	1364	677	945	857	646	4489
Number of words MT+Self-PE+REV (EN)	1389	685	934	859	611	4478

Table 2: Word counts per experimental condition

As stated earlier, we engaged five medical practitioners in our experiments and asked them to draft a paper or a section of a paper of around 750 words, or to send us a paper they had already written in Spanish and intended to translate into English. Table 2 offers a general overview of the number of words they originally wrote in Spanish as well as the breakdown of the word counts after each stage in the experiment.

5.4. Types of Edits

We aimed at identifying the type of edits that medical practitioners make when they engage in the self-post-editing process without any prior training in MT or post-editing using the typology outlined in Section 4.2.3.

Figure 1 shows the edit rates per participant.[5] Edits provoked by other edits were counted as a single edit in our analysis, as they would not have happened, if the first edit had not been made.

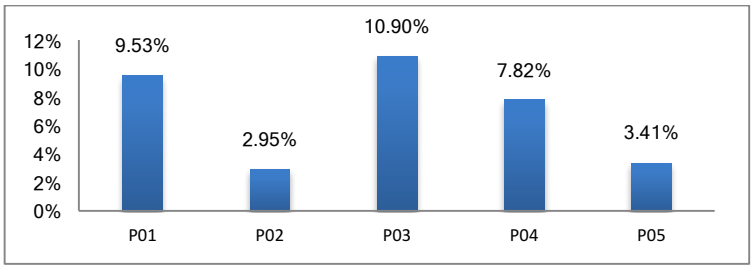


Figure 1: Edit rates per participant

As indicated in Figure 1, P03 was the author who has the highest edit rate, followed by P01 and P04. A potential explanation for this may be related to their English level. Both P02 and P05 had a B1 level of English according to the test (P05 was actually between B1 and B2), and also reported a B1 in their self-assessment. The other participants, on the other hand, had a B2 or C1 level (according to the test, P04 was between B2 and C1, and P01 between C1 and C2). It could therefore be the case, that a lower level of English hampers the ability to post-edit. This was also hinted at by P02 who declared that the MT output outperformed his/her level of English.

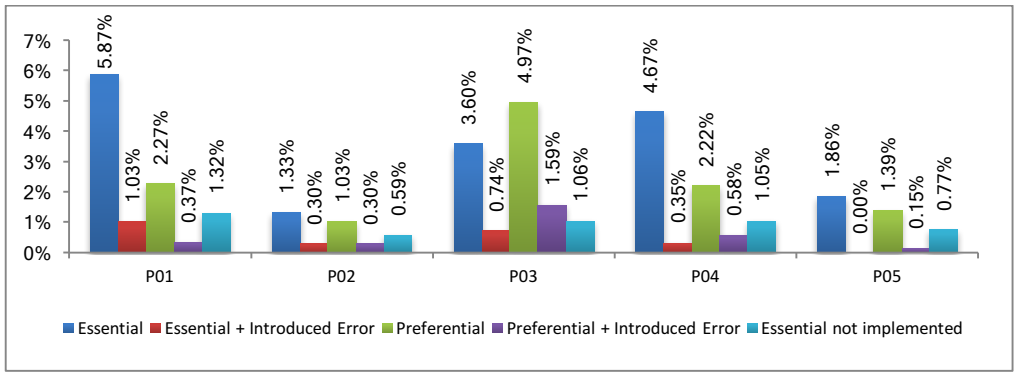


Figure 2: Types of edit per participant measured in edit rates

Our second research question was: "*Without any training in MT or post-editing, what type of edits do medical practitioners make when they write in Spanish and then machine translate into English and self-post-edit?*". If we break down the types of edits made (cf. Figure 2), our analysis shows that medical practitioners are able to identify and implement essential edits during post-editing without any prior training. P01 and P04, the two participants with the highest edit rates as per Figure 1, are precisely the two participants who made the highest rate of essential edits (5.87% and 4.67% respectively). However, P01 was also the participant who had the highest rate of essential edits not implemented (1.32%), followed by P03 and P04 (1.06% and 1.05% respectively). This additionally replies to our related question, "*Are essential edits implemented or ignored?*".

An interesting observation during the annotation was that in many cases the essential edits in the text were related to spelling and grammar, highlighting the importance of using

[5] By "edit rate" here we mean the number of edits implemented per 100 words of raw MT output, expressed as a percentage

spelling and grammar checkers as writing aids for non-native speakers of English. It was also surprising to see how Google Translate performed well even with noisy text, as the original texts in Spanish contained some grammar and spelling errors. For example, *típica* (typical) was spelt *tìpica*; *nuestro* (our), *nuesto*, and *excluida* (excluded), *excluída*, and yet the MT system translated them correctly. In any case, it does seem to hold true that those participants with a higher level of English identified and implemented more essential edits than those with a lower English level.

With regard to the subquestion, "*How much non-essential (or preferential) editing is carried out?*", we observed that again there is a tendency to implement preferential edits too. In our small cohort, only one participant (P03) implemented more preferential than essential edits. The extent of edits implemented varies however per individual, which has also been observed among professional translators who post-edit (e.g. de Almeida, 2013; Bundgaard, 2017). We found, for instance, that participants had different preferences regarding the use of technical versus colloquial terms, which was reflected in their edits. For example, P02 changed 'axillae' to 'armpits', whereas P05 seemed to prefer a more formal final text and changed expressions such as 'hospital stay' to 'hospitalization time'.

Our last related question addressed *whether errors are introduced via self-post-editing*. As with professional translators, medical practitioners also introduced some errors while editing, though the rates are relatively low. P01 was the participant who introduced the highest rate of errors when making essential edits (1.03%), followed by P03 (0.74%). P03 was also the participant that introduced more errors when implementing preferential edits (1.59%), and as a result the one who had the highest rate of introduced errors overall (2.33%). Further investigation is needed to identify the nature of the errors introduced and determine whether they could have been avoided (e.g. by means of spell and grammar checkers in the case of introduced typos). However, this might also have to do with the *need* for edits: within the medical domain, there exist several sub-domains and genres. This raises a new research question worth investigating in our future work: *Did Google Translate perform better in some sub-domains than others?*

5.5. Professional Proofreading

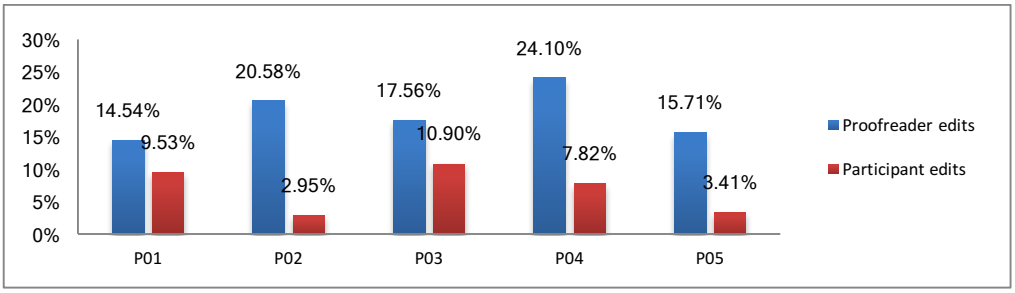


Figure 3: Edit rates per participant (proofreader vs. participants)

We subsequently analysed the edits made by the professional proofreader on the texts already self-post-edited by our participants. Figure 3 shows the overall edit rates per participant. The edit rates of each medical practitioner are provided to allow for an easier comparison. As may be observed, the professional revision of the texts resulted in a higher edit rate in all cases, with P04's text being the one that recorded the highest edit rate, followed by P02's. This is an interesting finding, as P04 was precisely one of the participants with a higher level of English, which suggests that the English level is not necessarily correlated with the post-editing ability. In some cases, e.g. P02 and P05, the proofreader introduced a significantly higher number of edits than

the participant, leading to edit rates that are more than five times those of the participants. An obvious question is whether or not the proofreader edits were indeed necessary or, were rather preferential and not strictly required. This is particularly relevant, because, as mentioned earlier, we asked the proofreader to focus on a mere surface revision of the text.

Similar to what we did with the texts undergoing self-PE we also annotated all edits per type of edit. Figure 4 summarises the edit rates of the proofreader classified by type. It is striking how in some cases the rate of preferential edits made was as high as that for essential edits (P04), or even higher (P01, P03 and P05). Only in the case of P01 was the rate of preferential edits lower than that of essential edits. Surprisingly, the proofreader also introduced some errors in the text while implementing edits. It is interesting to note that the number of introduced errors is higher in the case of preferential edits than in the case of essential ones. In some cases, the error introduced may have been caused by the use of "track changes" (e.g. when correcting the spelling of "pacient", she accidentally deleted the space between the word being corrected and the next: "the ~~pacient~~patientwas urgently…"), but the degree to which this influenced the editing process is difficult to gauge. In the case of P03, some of the errors introduced had to do with the bibliographical style, as the medical practitioner had opted for references between parenthesis and our proofreader changed them to superscript.

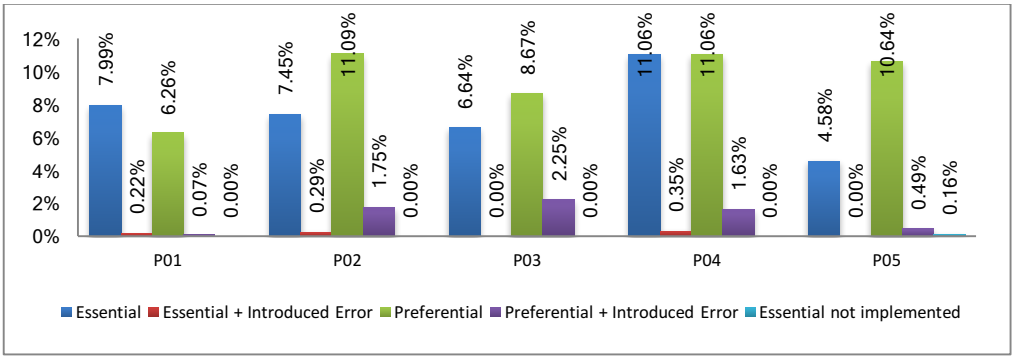


Figure 4: Types of edits by proofreader for each text

This analysis of the professional proofreader edits allows us to answer our third research question: "*How much editing is required by a professional proofreader on top of the post-edited documents and what type of edits are implemented?*". Indeed, the proofreader implemented a considerable number of edits. However, according to our typology, the proofreader also implemented a surprising number of preferential edits and even introduced some errors during the proofing process, though these were low in number. This seems to indicate that the proofreader is still required and that the post-editing process by our small cohort of medical practitioners did not render the text to a level of quality such that the proofreader thought that it required little to no editing. As an aside, this question also arises in professional practice and the general practice is still to have a revision after post-editing, which indicates that our findings would not be out of line with normal machine translation workflows.

Although we are not doing a comparison here between the number of required edits after post-editing versus the number of required edits to texts directly written in EFL, in a previous experiment we observed that the proofreader implemented more or less an equal number of edits on text that had been post-edited and text that had been written in EFL (O'Brien et al., forthcoming; Goulet et al., forthcoming). In future work it would be interesting to test if the same findings can be replicated in the medical domain.

6. Conclusion and Future Work

Here, we have reported on a small experiment seeking to explore the usefulness of MT as a writing aid for Spanish medical practitioners that need to publish their work in English. Thanks to the general survey we conducted, we found that Spanish-speaking medical practitioners are already using MT as a writing aid. However, they also showed mixed feelings about its usefulness. Some of the main criticisms had to do with the literalness of the MT output, incorrect use of synonyms, grammar and the lack of terminology. This raises a question as to whether domain-tuned MT engines might solve some of these issues.

Our analysis revealed that medical practitioners perform both essential and preferential edits (3.90% and 2.52% respectively and on average for all participants), and that the professional proofreader hired to proof the self-post-edited texts written by our participants also implemented both types of changes (7.75% and 9.02%). Surprisingly, in the case of the proofreader the rate of preferential edits was higher than that of the essential ones. This seems to agree with what has been observed in professional translation workflows, as demonstrated by Bundgaard (2017). In an investigation of professional translators' edits during the "checking phase" of translations (translators checking their own work) Bundgaard (2017: 205) found the rate of preferential edits to be 43% on average for one text and 66% on average for a second text in her experiment, i.e. of all edits implemented for one text, 43% of them were deemed to be 'preferential'. Bundgaard was also using de Almeida's typology for assessing essential and preferential edits. Bundgaard (2017: 225) also measured the number of essential and preferential edits implemented by a third party during a 'review phase' (an independent translator checking another translator's work) and these ranged from 37% on average for one text and 60% for the second text.

Our analysis of the edits made by medical practitioners and the subsequent engagement of a professional proofreader additionally sought to answer whether medical practitioners would be in a position to carry out self-post-editing without any prior training and whether they were able to achieve an acceptable quality text with MT. Overall, without training, these experts can implement essential edits, but they also implement preferential edits and introduce errors. This raises the question as to whether further training and practice would make medical practitioners better post-editors.

At the same time, the proofreader's intervention demonstrated that an important number of essential edits had not been implemented by the medical practitioners. Yet, the proofreader also implemented a high proportion of preferential edits, according to our typology. It is still to be determined whether the texts produced by our participants would have actually been considered acceptable for publications or presentations in medical conferences where non-native speakers of English also present their work. In future work we plan to engage native speakers to assess this. We may consider, for example, asking them to rank the post-edited version against the proofread version to ascertain whether, and to what extent, the proofread version is acceptable as well as whether, and to what extent, it is superior to the post-edited version.

Similarly to what we did in Goulet et al. (forthcoming), we also plan to carry out a second round of annotation in which we will annotate the type of edit made (insert, delete, move, replace), the type of language unit affected in each case (noun, verb, preposition, etc.), and the linguistic dimension involved (morphology, syntax, semantics, etc.). This will allow us to analyse the edits further, make comparisons across the edits made by the experts and the professional proofreader, and determine whether automatic post-editing could be used to enhance the text prior to the self-post-editing process.

To sum up, our results, while demonstrating that the medical practitioners were capable of post-editing their own texts to some degree, do not seem to indicate that they could produce

their final drafts of scientific papers under the current experimental setup, i.e. with a generic engine, no automated post-editing rules and no intervention by a proofreader. However, the small cohort engaged in our experiment (five participants) does not allow us to draw a general conclusion. This experiment helped us to identify several avenues to improve our experimental setup and we will endeavour to address the issues identified and answer these new questions in our future work.

Acknowledgements

This research is supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 713567, and Science Foundation Ireland in the ADAPT Centre (Grant 13/RC/2106) (www.adaptcentre.ie) at Dublin City University.

References

Benfield J. R., and Feak, C. B. (2006). How authors can cope with the burden of English as an international language. *Chest*, 129(6), 1728-1730.

Bennett, K. (2013). English as a lingua franca in academia. *The Interpreter and Translator Trainer* 7/2, 169-193.

Bennett, K. (2014). The political and economic infrastructure of academic practice: the 'semi-periphery' as a category for social and linguistic analysis. In Bennett, K. (dir.), *The semi-periphery of academic writing: Discourses, communities and practices*. London: Palgrave Macmillan, 1-12.

Bennett, K. (2015). Towards an epistemological monoculture: mechanisms of epistemicide in European research publication. In Plo, R. and C. Pérez-Llantada (dir.), *English as an academic and research language (English in Europe Vol. 2)*. De Gruyter Mouton: Berlin, 9-35.

Breuer, E. O. (2015). *First language versus foreign language. Fluency, errors and revision processes in foreign language academic writing*. Frankfurt am Main: Peter Lang Edition.

Bundgaard, K. (2017). (*Post-)Editing - A workplace study of translator-computer interaction at Textminded Danmark A/S*. PhD Thesis. Aarhus University, Denmark.

De Almeida, Giselle (2013) *Translating the post-editor: an investigation of post-editing changes and correlations with professional experience across two Romance languages*, Unpublished PhD Thesis, Dublin City University.

Garcia, I., and Pena, M. I. (2011). Machine translation-assisted language learning: Writing for beginners. *Computer Assisted Language Learning*, 24(5), 471-487. doi: 10.1080/09588221.2011.582687.

Goulet, M.J., Simard, M., Parra Escartín, C. and O'Brien, S (forthcoming). *La traduction automatique comme outil d'aide à la rédaction scientifique en anglais langue seconde : résultats d'une étude exploratoire sur la qualité linguistique*. Anglais de Spécialité (ASp).

Lillis, T. and Curry, M. J. (2010). *Academic writing in a global context: The politics and practices of publishing in English*. London and New York: Routledge.

LISA (Localization Industry Standards Association). (2004). *LISA QA Model 3.0. Product Documentation.* Féchy, Switzerland: LISA.

Niño, A. (2008). Evaluating the use of machine translation post-editing in the foreign language class. *Computer Assisted Language Learning*, 21(1), 29-49.

NIST (National Institute of Standards and Technology) Information Access Division/Speech Group and Linguistic Data Consortium (LDC). (2007). *Post Editing Guidelines for GALE Machine Translation Evaluation*. Version 3.0.2.

O'Brien, S., Simard, M. and Goulet, M. J. (forthcoming, 2018). Machine Translation and Self-Post-Editing for Academic Writing Support: Quality Explorations. In: Moorkens, J., Castilho, S., Doherty, S. and Gaspari, F, Springer Series on Machine Translation. Springer.

O'Neill, E. M. (2012). *The effect of online translators on L2 writing in French*, PhD thesis, University of Illinois at Urbana-Champaign, USA. http://hdl.handle.net/2142/34317. Accessed May 8 2017.

Skelton, J. R. and Edwards, S. J. L. The Function of the Discussion Section in Academic Medical Writing. *BMJ : British Medical Journal* 320.7244 (2000): 1269–1270.

Willey, I., and Tanimoto, K. (2015). "We're drifting into strange territory here": What think-aloud protocols reveal about convenience editing. *Journal of Second Language Writing*, 27, 63-83, (2015). doi: 10.1016/j.jslw.2014.09.010.

A Multilingual Parallel Corpus for Improving Machine Translation on Southeast Asian Languages

Hai-Long Trieu, Le-Minh Nguyen {trieulh, nguyenml}@jaist.ac.jp
Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan

Abstract

Current machine translation systems require large bilingual corpora for training data. With large bilingual corpora, phrase-based and neural-based methods can achieve state-of-the-art performance. Nevertheless, such large bilingual corpora are unavailable for most language pairs called low-resource languages, which causes a bottleneck for the development of machine translation on such languages. For Southeast Asian region, there is a large population with more than five hundred millions people and several languages that can be used popularly in the world, but there are few parallel data for such language pairs. In this work, we built a multilingual parallel corpus for several Southeast Asian languages. Wikipedia articles' titles and inter-language link records were used to extract parallel titles. Parallel articles were collected based on the parallel titles. For each article pair, parallel sentences were extracted based on a length-based and word correspondences sentence alignment method. A multilingual parallel corpus were built with more than 1.1 million parallel sentences of ten language pairs of Indonesian, Malay, Filipino, Vietnamese and the languages paired with English. Experiments were conducted on the Asian Language Treebank corpus and showed the promising performance. Additionally, the corpus was utilized for the IWSLT 2015 machine translation shared task on English-Vietnamese and achieved a significant improvement with +1.7 BLEU point on phrase-based systems and +4.5 BLEU point on a state-of-the-art neural-based system. The corpus can be used to improve machine translation and enhance the development of machine translation on the low-resource Southeast Asian languages.

1 Introduction

Current machine translation (MT) systems require large bilingual corpora for training data. With large bilingual corpora up to millions of parallel sentences, MT systems achieve the state-of-the-art performance on both phrase-based (Bojar et al., 2013) and neural-based (Sennrich et al., 2016a) methods. Such large bilingual corpora are available on several language pairs such as English-German, English-French, Czech-English, Chinese-English. For low-resource language pairs, which are most of languages in the world (Irvine, 2013; Wang et al., 2016), there are only small bilingual corpora available. This causes a bottleneck for MT on such language pairs.

In order to overcome the problem, previous works have made efforts in building bilingual corpora from webs such as in (Utiyama and Isahara, 2003; Li and Liu, 2008; Cettolo et al., 2012). The parallel corpora can be extracted from comparable data such as Wikipedia ((Ştefănescu and Ion, 2013; Chu et al., 2015). The previous work contributed for building bilingual corpora automatically for several low-resource language pairs. For Southeast Asian

languages, there are few bilingual corpora on the languages although there are a high population with more than five hundred millions of people, and there are several languages that can be used popularly in the world such as Indonesian (ranked 12), Vietnamese (ranked 17) as the most popularly used languages (Weber, 2008). This causes an issue for the development of machine translation on the language pairs.

In this work, we built a multilingual parallel corpus to improve machine translation for Southeast Asian languages, which there is no large bilingual corpora. Parallel titles of Wikipedia articles were extracted based on the articles' titles and inter-language link records from the Wikipedia database. Parallel articles were collected based on the parallel titles. Then, parallel sentences were aligned based on a sentence alignment method that is the combination of length-based and word correspondences. A multilingual parallel corpus was built for several low-resource Southeast Asian languages that included more than 1.1 million parallel sentences of ten language pairs between Indonesian, Filipino, Malay, Vietnamese and these languages paired with English. Experiments of machine translation were conducted on the Asian Language Treebank corpus (Thu et al., 2016). Experimental results showed that using the extracted corpus to build machine translation systems can achieve promising results although there is no direct bilingual corpora. Furthermore, experiments were conducted on the IWSLT 2015 machine translation shared task (Cettolo et al., 2015) using the extracted corpus for English-Vietnamese trained on phrase-based and neural-based machine translation systems. Experimental results showed that using the extracted corpus achieved significant improvement in both phrase-based systems and neural-based systems. The corpus can be used to improve machine translation performance and enhance the development of machine translation for the Southeast Asian languages. We released the extracted corpus and the code to build the corpus, which are available at the repository.[1]

We briefly discuss related work in Section 2. The procedures to built the corpus are described in detail in Section 3. The statistics of the extracted corpus are presented in Section 4. In order to effectively utilize the corpus, we present several strategies to exploit the corpus for machine translation in Section 5. Experiments are described in Section 6 to evaluate and utilize the corpus. Conclusions are drawn in Section 7.

2 Related Work

Building parallel corpora from webs has been exploited in a long period. One of the first work can be presented in Resnik (1999). In order to extract parallel documents from webs, Li and Liu (2008) used the similarity of the URL and page content. Utiyama and Isahara (2003) used matching documents to build parallel data. Meanwhile, Koehn (2005) used manual involvement for building a multilingual parallel corpus. In the work of Cettolo et al. (2012), a multilingual corpus was built from subtitles of the TED talks website.

For collecting parallel data from Wikipedia, the task has been investigated in some previous work. In the work of Kim et al. (2012), parallel sentences are extracted from Wikipedia for the task of multilingual named entity recognition. In Ştefănescu and Ion (2013), parallel corpora are extracted from Wikipedia for English, German, and Spanish. A recent work proposed by Chu et al. (2015) extracts parallel sentences before using an SVM classifier to filter the sentences using some features.

For the Southeast Asian languages, there are few bilingual corpora. A multilingual parallel corpus was built manually in Thu et al. (2016). The corpus is a valuable resource for the languages. Nevertheless, because the corpus is still small with only 20,000 multilingual sentences, and manually building parallel corpora requires many cost of human annotators, automatically extracting large bilingual corpora becomes an essential task for the development of natural lan-

[1] https://github.com/nguyenlab/Multi-Wiki

guage processing for the languages including cross-language tasks like machine translation. In our work, a multilingual parallel corpus of several Southeast Asian languages was built. The corpus was built based on Wikipedia's parallel articles that were collected from the articles' title and inter-language link records. Parallel sentences were extracted based on the powerful sentence alignment algorithm (Moore, 2002). The corpus was utilized for improving machine translation on the Southeast Asian low-resource languages, in which there has been no work investigated on this task to our best knowledge.

3 Methods

Wikipedia is a large resource that contains a number of articles in many languages in the world. The freely accessible resource is a kind of comparable data in which many articles are in the same domain in different languages. We can exploit this resource to build bilingual corpora, especially for low-resource language pairs.

In order to build a bilingual corpus from Wikipedia, we first extracted parallel titles of Wikipedia articles. Then, pairs of articles were crawled based on the parallel titles. Finally, sentences in the article pairs were aligned to extract parallel sentences. We describe these steps in more detail in this section.

3.1 Extracting Parallel Titles

The content of Wikipedia can be obtained from their database dumps.[2] In order to extract parallel titles of Wikipedia articles, we used two resources for each language from the Wikipedia database dumps: the articles' titles and IDs in a particular language (ending with *-page.sql.gz*) and the interlanguage link records (file ends with *-langlinks.sql.gz*).

No.	Data	page (KB)	langlinks (KB)
1	en	1,477,861	280,617
2	vi	92,541	111,420
3	id	57,921	72,117
4	ms	21,791	56,173
5	fil	5,907	23,446

Table 1: Wikipedia database dumps' resources for extracting parallel titles; **page (KB)**: the size of the articles' IDs and their titles in the language; **langlinks (KB)**: the size of the interlanguage link records; we collected the resources for languages: **en** (English), **id** (Indonesian), **fil** (Filipino), **ms** (Malay), and **vi** (Vietnamese); we used the database that was *updated on 2017-01-20*.

We aim to build a multilingual parallel corpus for several low-resource Southeast Asian languages including Indonesian, Malay, Filipino, and Vietnamese, which there are few bilingual corpora. Furthermore, bilingual corpora of the languages paired with English are also important resources for further research including machine translation. Therefore, we collected the Wikipedia database dumps of the five languages: English, Indonesian, Malay, Filipino, and Vietnamese. Table 1 presents the Wikipedia database dumps that we used to extract parallel titles. The English database contains a much larger information in both the articles' titles and the interlanguage link records. Meanwhile, the Filipino database is much smaller, that affects the number of extracted parallel titles as well as final extracted parallel sentences. The extracted parallel titles are presented in Table 2.

[2] https://dumps.wikimedia.org/backup-index.html

No.	Data	Title pairs	Crawled Src Art.	Crawled Trg Art.	Art. Pairs	Src Sent.	Trg Sent.
1	en-id	198,629	197,220	190,954	150,759	4,646,453	990,661
2	en-fil	52,749	51,698	51,157	50,021	3,428,599	367,276
3	en-ms	204,833	201,688	199,950	160,709	2,158,726	320,624
4	en-vi	452,415	433,124	436,488	420,919	12,130,133	3,831,948
5	id-fil	30,313	29,961	24,946	22,760	502,457	254,216
6	id-ms	98,305	88,028	89,936	68,676	452,604	403,807
7	id-vi	159,247	149,974	128,530	121,673	1,201,848	1,878,855
8	fil-ms	25,231	21,856	25,023	21,135	202,851	243,361
9	fil-vi	36,186	30,540	35,625	28,830	267,453	723,155
10	ms-vi	133,651	118,647	116,620	105,692	560,042	1,256,468

Table 2: Extracted and processed data from parallel titles; **Crawled Src Art. (Crawled Trg Art.)**: the number of crawled source (target) articles using the title pairs for each language pair; **Art. Pairs**: the number of parallel articles processed after crawling; **Src Sent. (Trg Sent.)**: the number of source (target) sentences in the article pairs after preprocessing (removing noisy characters, empty lines, sentence splitting, word tokenization).

3.2 Collecting and Preprocessing Parallel Articles

After parallel titles of Wikipedia articles were extracted, we collected the article pairs using the parallel titles. We implemented a Java crawler for collecting the articles. The collected data set was then carefully processed in hierarchical steps from articles to sentences, then to word levels. First, noisy characters were removed from the articles. Then, for each article, sentences in paragraphs were splitted so that there is one sentence per line. For each sentence, words were tokenized that separated from punctuations. The sentence and word tokenization steps were conducted using the Moses scripts.[3]

As described in Table 2, using the title pairs, we obtained a high ratio of crawled articles. For instance, using 198k title pairs of English-Indonesian, we crawled 197k English articles and 190k Indonesian articles successfully, which there existed the article based on a title. This issue is emphasized because sometimes there is no existed article given a title that will show an error in crawling. For the case of Indonesian-Vietnamese, although there was 159k extracted parallel titles, we obtained 128k Vietnamese articles, which there were more than 30k error or inexistent articles given the set of titles.

3.3 Aligning Parallel Sentences

Sentence alignment is an essential task in building parallel corpora. In the three main approaches in sentence alignment: length-based which is based on the number of words or characters (Brown et al., 1991; Gale and Church, 1993), word-based which is based on word correspondences (Kay and Röscheisen, 1993; Chen, 1993; Wu, 1994; Melamed, 1996; Ma, 2006), and the combination of length-based and word-based (Moore, 2002; Varga et al., 2007), the hybrid method of Moore (2002) achieved the best performance compared with other sentence alignment approaches as the evaluation of Singh and Husain (2005). In our work, for each parallel article pair, we aligned sentences using the Microsoft bilingual sentence aligner (Moore, 2002). There are several reasons to adapt the hybrid method for aligning parallel sentences in this task. First, the length-based method has been applied successfully in close languages such as English-French; however, the languages in the Southeast Asian including Indonesian, Malay,

[3] https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer

Vietnamese, Filipino, and the languages paired with English are not closed languages exception for the Indonesian-Malay. Second, since the Wikipedia bilingual articles are the kind of comparable data, it varies greatly in terms of the number of sentences in bilingual articles and the number of words in sentence pairs. Therefore, we adapted the hybrid method that combines the length-based and word correspondences to extract the parallel corpus.

Let l_s and l_t be the lengths of source and target sentences, respectively. Then, l_s and l_t varies according to Poisson distribution as follows:

$$P(l_t|l_s) = exp^{-l_t r}\frac{(l_s r)^{l_t}}{l_t!} \quad (1)$$

Where *r* is the ratio of the mean length of target sentences to the mean length of source sentences. As shown in the method of Moore (2002), the length-based phase based on the Poisson distribution

Sentence pairs extracted from the length-based phase are then used to train IBM Model 1 (Brown et al., 1993) to build a bilingual dictionary. The dictionary was then combined with the length-based phase to produce final alignments, which are described as follows:

$$P(s,t) = \frac{P_{1-1}(l_s,l_t)}{(l_s+1)^{l_t}}(\prod_{j=1}^{l_t}\sum_{i=0}^{l_s} tr(t_j|s_i))(\sum_{i=1}^{l_e} f_u(e_i)) \quad (2)$$

Where: $tr(t_j|s_i)$ is the probability of the word pair $(t_j|s_i)$ trained by IBM Model 1; f_u is the observed relative unigram frequency of the word in the text in the corresponding language.

Challenges in aligning Wikipedia articles As we discussed above, the Wikipedia article pairs greatly vary in terms of sentence length in the article pairs because of this kind of comparable data. Furthermore, in some article pairs, the articles in two languages even contain many differences in content, priorities, interests, and bias of the authors, groups or countries involved, etc. Such differences cause many challenges for aligning Wikipedia articles to create a parallel corpus. For our first effort in building this corpus, we used the hybrid sentence alignment method to extract sentence pairs for the first version of this corpus without any strategy to filter or extract parallel sentences in dealing with these challenges. We plan to conduct further analysis as well as strategies to deal with the challenges and improve the quality of this corpus in future work. A method proposed in Munteanu and Marcu (2006) can be utilized for this task, in which parallel sub-sentential fragments are extracted from comparable data.

4 Extracted Corpus

We obtained a multilingual parallel corpus of ten language pairs, which are among Southeast Asian languages and the languages paired with English as described in Table 3. In totally, the corpus contains a huge number of parallel sentences up to more than 1.1 million sentence pairs which can be valuable when there is no available bilingual corpora for almost such language pairs. Large bilingual corpora can be extracted such as: English-Vietnamese (408k parallel sentences), Indonesian-English (234k parallel sentences). However, because of the smaller number of the input parallel articles for several language pairs, a much smaller number of parallel sentences were extracted like Indonesian-Filipino (9k) and Filipino-English (22k).

Furthermore, we extracted monolingual data sets for the languages: Indonesian, Malay, Filipino, and Vietnamese, which are almost publicly unavailable. The data sets are described in Table 4. Large monolingual data sets were obtained such as Indonesian (3.1 million sentences), Malay (1.5 million sentences), and Vietnamese (up to 7.6 million sentences). The data sets are useful for such low-resource languages such as training language models and other tasks.

No.	Data	Sent. Pairs	Src Words	Trg Words	Src Vocab.	Trg Vocab.
1	en-id	234,380	4,648,359	4,359,976	208,920	209,859
2	en-fil	22,758	447,719	399,058	42,670	44,809
3	en-ms	198,087	3,273,943	3,221,738	156,806	148,133
4	en-vi	408,552	7,229,963	8,373,549	274,178	222,068
5	id-fil	9,952	132,097	172,363	18,531	19,737
6	id-ms	83,557	1,464,506	1,447,247	87,240	92,126
7	id-vi	76,863	1,014,351	1,136,710	67,211	57,788
8	fil-ms	4,919	78,729	66,324	10,184	10,671
9	fil-vi	10,418	141,135	151,086	15,641	13,071
10	ms-vi	65,177	928,205	896,784	60,574	52,673
	Total	1,114,663	–	–	–	–

Table 3: Extracted Southeast Asian multilingual parallel corpus

Data set	Sentences	Vocab.	Size (KB)
id	3,147,570	917,861	369
fil	1,034,215	252,565	113
ms	1,527,834	599,396	172
vi	7,690,426	936,137	1,033

Table 4: Monolingual data sets

5 Domain Adaptation

The question now is that how can we utilize the corpus effectively. If there are existing bilingual corpora for the language pairs, which strategies we can use to combine and take advantage the full potential of the corpus. We discuss the issue of domain adaptation about the strategies to combine bilingual corpora in this section.

We assume that given a language pair, there exist a bilingual corpus called the *direct corpus*. The corpus extracted from Wikipedia can be used as an additional resource, called the *alignment corpus*. For phrase-based machine translation (Koehn et al., 2003), a bilingual corpus is used to train a phrase table. We used the *direct corpus* and the *alignment corpus* to generate two phrase tables called the *direct* and the *alignment* components. The two components were combined using the linear interpolation as described in Equation 3.

$$p(t|s) = \lambda_d p_d(t|s) + \lambda_a p_a(t|s) \tag{3}$$

where $p_d(t|s)$ and $p_a(t|s)$ stand for the translation probabilities of the *direct* and the *alignment* models, respectively; interpolation parameters: λ_d and λ_a (where $\lambda_d + \lambda_a$=1).

We adapted the linear interpolation (Sennrich, 2012), which is a robust method for a weighted combination of translation models. Specifically, we used two strategies called *tune* and *weights*.

- *tune*: a tuning set was used; λ_d and λ_a were calculated as the weights that minimize cross-entropy on the tuning set using the setting *combine_given_tuning_set* (Sennrich, 2012). [4]
- *weights*: The two translation models were first used for decoding the tuning set separately to generate two BLEU scores. Then, the interpolation weights were set using the ratios of the two BLEU scores using the setting *combine_given_weights* Sennrich (2012).

[4] https://github.com/moses-smt/mosesdecoder/tree/master/contrib/tmcombine

6 Experiments on Machine Translation

The parallel corpus extracted from Wikipedia was then used for training SMT models. We aim to exploit the data to improve SMT on low-resource languages.

6.1 SMT on the Asian Language Treebank Parallel Corpus

6.1.1 Training Data

We evaluate the corpus on SMT experiments. For development and testing data, we used the ALT corpus (Asian Language Treebank Parallel Corpus) Thu et al. (2016), this is a corpus including 20K multilingual sentences of English, Japanese, Indonesian, Filipino, Malay, Vietnamese, and some other Southeast Asia languages. We extracted the development and test sets from the ALT corpus: 2k sentence pairs for development sets, and 2k sentence pairs for test sets.

6.1.2 Training Details

We trained SMT models on the parallel corpus using the Moses toolkit (Koehn et al., 2007). The word alignment was trained using GIZA++ (Och and Ney, 2003) with the configuration *grow-diag-final-and*. A 5-gram language model of the target language was trained using KenLM (Heafield, 2011). For tuning, we used batch MIRA (Cherry and Foster, 2012). For evaluation, we used the BLEU scores (Papineni et al., 2002) based on the *multi-bleu.perl* script; the development sets, test sets, and scripts to calculate the BLEU scores are also available in the repository of this paper.

6.1.3 Results

Table 5 describes the experimental results on the development and test sets. It is noticeable that the SMT models trained on the bilingual data aligned from Wikipedia produced promising results.

No.	Language Pairs	Dev (L1-L2)	Test (L1-L2)	Dev (L2-L1)	Test (L2-L1)
1	en-id	30.56	28.87	30.14	29.01
2	en-fil	18.54	19.08	18.98	19.89
3	en-ms	29.85	33.23	28.87	23.82
4	en-vi	30.58	34.42	23.01	22.56
5	id-fil	11.36	11.04	9.58	9.77
6	id-ms	31.64	30.21	31.56	30.11
7	id-vi	21.85	22.42	17.41	17.45
8	fil-ms	7.43	8.02	8.70	9.27
9	fil-vi	5.97	6.69	6.45	7.15
10	ms-vi	15.51	18.12	11.96	13.88

Table 5: Experimental results on the development and test sets (BLEU); **Dev (L1-L2)**, **Test (L1-L2)**, **fil-ms**: the translation scores on the development (test) set of the translation from the first language (**L1(fil)**) to the second language (**L2 (ms)**) in the language pair **fil-ms**; **Dev (L2-L1)**, **Test (L2-L1)**, **fil-ms**: the translation on the development (test) set of the inverse translation (from **ms** to **fil**)

For the results on the development sets, we achieved promising results with high BLEU points such as: the Indonesian-Malay pairs (Indonesian-Malay 31.64 BLEU points, Malay-Indonesian 31.56 BLEU points). Similarly, several language pairs also showed high BLEU points such as: English-Vietnamese (30.58 and 23.01 BLEU points), English-Malay (29.85

and 28.87 BLEU points), English-Indonesian (30.56 and 30.14 BLEU points), and Indonesian-Vietnamese (21.85 and 17.41 BLEU points). The language pairs which showed high scores contain a large number of sentences, for instance English-Vietnamese (408k sentence pairs), English-Indonesian (234k sentence pairs), and English-Malay (198k sentence pairs). Nevertheless, since the small number of the extracted corpus on several languages paired with Filipino such as Indonesian-Filipino (9.9k sentence pairs), Malay-Filipino (21.1k sentence pairs), and Vietnamese-Filipino (10.4k sentence pairs), the experimental results showed much lower performance than other language pairs: Indonesian-Filipino (11.36 and 9.58 BLEU points), Malay-Filipino (8.70 and 7.43 BLEU points), and Vietnamese-Filipino (6.45 and 5.97 BLEU points).

Similarly, for the experimental results on the test sets, the language pairs with large bilingual corpora achieved high performance: English-Indonesian (28.87 and 29.01 BLEU points), English-Malay (33.23 and 23.82 BLEU points), English-Vietnamese (34.42 and 22.56 BLEU points). The situation of languages paired Filipino showed the much lower performance: Indonesian-Filipino (11.04 and 9.77 BLEU points), Malay-Filipino (9.27 and 8.02 BLEU points), and Vietnamese-Filipino (7.15 and 6.69 BLEU points).

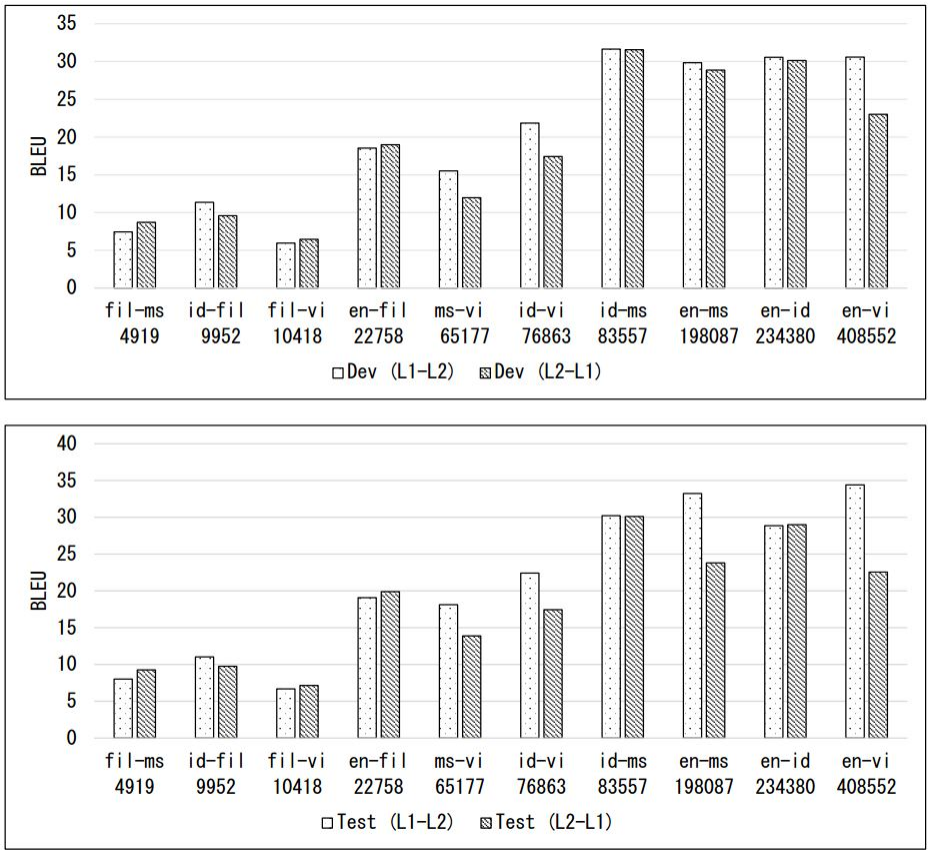


Figure 1: Experimental results on the development and test sets; the corpus's size is presented for each language pair (**fil-ms 4919**: the Filipino-Malay corpus with 4,919 parallel sentences)

Figure 1 presents experimental results on the development sets (test sets) that vary in

several aspects: the translation directions (L1-L2, L2-L1), the corpus's size, and the language pairs. There are several interesting findings from the charts. First, the bigger the corpus's size, the higher the BLEU scores. We sorted the corpus's size increasingly from the left to right. For instance, since the corpora' sizes of language pairs such as Filipino-Malay (4.9k), Indonesian-Filipino (9.9k), and Filipino-Vietnamese (10.4k) are much smaller than that of the language pairs such as Indonesian-Malay (83.5k), English-Indonesian (234k), English-Vietnamese (408k), the BLEU scores also show the correlation of the two language-pair groups: Filipino-Malay, Indonesian-Filipino, Filipino-Vietnamese ($<$10 or $\approx$ 10 BLEU points); Indonesian-Malay, English-Indonesian, English-Vietnamese ($\approx$ 25-30 BLEU points). Second, in the aspect of the translation directions (L1-L2, L2-L1), the scores of the two translations in each language pair are mostly similar to each other in most cases, for instance: English-Indonesian (30.56 and 30.14 BLEU points in the two translation directions on the development set, 28.87 and 29.01 on the test set), Indonesian-Malay (31.64 and 31.56 BLEU points on the development set, 30.21 and 30.11 on the test set). Nevertheless, for Vietnamese, the translation scores from a language to Vietnamese are much higher than the translation scores from Vietnamese to that language in most cases, for instance: Malay-Vietnamese (15.51 BLEU point (ms-vi) vs. 11.96 (vi-ms) on the development set, 18.12 (ms-vi) vs. 13.88 (vi-ms) on the test set), Indonesian-Vietnamese (21.85 vs. 17.41 BLEU points on the development set, 22.42 vs. 17.45 BLEU points on the test set), and English-Vietnamese (30.58 vs. 23.01 BLEU points on the development set, 34.42 vs. 22.56 BLEU points on the test set). This problem of the unbalance scores between the two translation directions of a language paired with Vietnamese as well as other language pairs should be further investigated.

6.2 Evaluation on the IWSLT 2015 Machine Translation Shared Task

In this section, we evaluated the extracted corpus on the IWSLT 2015 machine translation shared task on English-Vietnamese. We aim to evaluate whether the *Wikipedia* corpus can improve some baseline systems on the shared task. In addition, we conducted various experiments of the domain adaptation strategies, statistical machine translation, and neural machine translation using the *Wikipedia* corpus to explore optimal strategies in exploiting the corpus.

6.2.1 Training Data

Data	Sentences	Src Words	Trg Words	Src Vocab.	Trg Vocab.
constrained	131,019	2,534,498	2,373,965	50,118	54,565
unconstrained	456,350	8,485,112	8,132,913	114,161	124,846
constrained+Wikipedia	538,981	9,710,389	9,017,601	288,785	345,839
unconstrained+Wikipedia	864,312	15,661,003	14,776,549	338,424	403,581
tst2012	1,581	28,773	27,101	3,713	3,958
tst2013	1,304	28,036	27,264	3,918	4,316
tst2015	1,080	20,844	19,951	3,175	3,528

Table 6: Data sets on the IWSLT 2015 experiments; **Src Words (Trg Words)**: the number of words in the source (target) side of the corpus; **Src Vocab. (Trg Vocab.)**: the vocabulary size in the source (target) side of the corpus

We used the data sets provided by the International Workshop on Spoken Language Translation (IWSLT 2015) machine translation shared task (Cettolo et al., 2015), which include three data sets of the training, development, and test sets extracted from subtitles of TED talks.[5] For

[5]https://www.ted.com/talks

the training data, the data set called the *constrained* data of 131k parallel sentences. The workshop provided data sets for development and test sets: *tst2012*, *tst2013*, and *tst2015*. In all experiments, we used the *tst2012* for the development set, the *tst2013* and *tst2015* for the test sets.

In addition, we used two other data sets for training data: the corpus of National project VLSP (Vietnamese Language and Speech Processing)[6] and the EVBCorpus (Ngo et al., 2013). The two data sets were merged with the *constrained* data to obtain a large training data set called the *unconstrained* data. The training, development, and test sets are described in Table 6.

6.2.2 Training Details

We trained translation systems using two methods: SMT and NMT.

Statistical Machine Translation In order to train SMT models, we used the well-known Moses toolkit (Koehn et al., 2007). The GIZA++ (Och and Ney, 2003) was used to train word alignment. For language model, we used KenLM (Heafield, 2011) to train 5-gram language models on the target side (Vietnamese) of the training data sets. The parameters were tuned using batch MIRA (Cherry and Foster, 2012). BLEU (Papineni et al., 2002) was used as the metric for evaluation.

Neural Machine Translation In our work, we based on the model of Sennrich et al. (2016a), which are encoder-decoder networks with an attention mechanism (Bahdanau et al., 2015). For NMT model, we adopted the attentional encoder-decoder networks combined with byte-pair encoding (Sennrich et al., 2016a). In our experiments, we set the word embedding size 500, and hidden layers size of 1024. Sentences are filtered with the maximum length of 50 words. The minibatches size is set to 60. The models were trained with the optimizer Adadelta (Zeiler, 2012). The models were validated each 3000 minibatches based on the BLEU scores on development sets. We saved the models for each 6000 minibatches. For decoding, we used beam search with the beam size of 12. We trained NMT models on an Nvidia GRID K520 GPU.

6.2.3 Results

Model	**tst2012**	**tst2013**	**tst2015**
Wikipedia	18.40	22.06	20.34
constrained	24.72	27.31	25.47
constrained+Wikipedia	24.78	27.89	26.69
constrained*Wikipedia (tune)	24.65	28.05	27.00
constrained*Wikipedia (weights)	**24.95 (+0.23)**	**28.51 (+1.20)**	**27.21 (+1.74)**
unconstrained	34.42	27.19	25.41
unconstrained+Wikipedia	33.88	27.28	26.36
unconstrained*Wikipedia (tune)	34.44	27.55	26.68
unconstrained*Wikipedia (weights)	**34.73 (+0.31)**	**28.04 (+0.85)**	**26.78 (+1.37)**

Table 7: Experimental results using phrase-based statistical machine translation; *constrained+Wikipedia*: the *constrained* data was merged with the *Wikipedia* corpus; *unconstrained*Wikipedia*: interpolation of the two models; *tune, weights*: the two interpolation settings; the **bold** indicates the best results for each setup

SMT results Table 7 presents experimental results using SMT models. Using the *Wikipedia* corpus, we achieved promising results: 18.40 BLEU point (*tst2012*), 22.06 (*tst2013*), and 20.34 (*tst2015*). When the *Wikipedia* corpus was merged with the *constrained* data for training data,

[6] http://vlsp.vietlp.org:8080/demo/?page=home

a significant improvement was achieved especially on the *tst2015* (26.69 BLEU point, which improved 1.22 BLEU point from the model using the *constrained* data). Nevertheless, the domain adaptation strategies show even better performance than the merging setting, in which the *weights* setting model obtained the best performance with +1.74 BLEU point improvement on the *tst2015*.

NMT results The NMT results are described in Table 8. From the experimental results, we can observe that the systems obtain the higher scores when the size of training data sets increase (from the *Wikipedia*, *constrained*, *unconstrained* to the merging in which the *unconstrained* data was merged with the *Wikipedia* corpus). It is interesting to note that using the *Wikipedia* corpus to enhance the translation systems trained on existed data sets based on NMT achieved the significant improvement up to +4.51 BLEU points on the *tst2015*.

Model	**tst2012**	**tst2013**	**tst2015**
constrained	20.21	23.59	17.27
Wikipedia	15.29	18.43	17.58
unconstrained	24.05	26.71	22.30
unconstrained+Wikipedia	**25.29 (+1.24)**	**28.93 (+2.21)**	**26.81 (+4.51)**

Table 8: Experimental results on neural machine translation (NMT) ; the **bold** indicates the best results for each setup

A work that enhanced neural machine translation using additional data is presented in Sennrich et al. (2016b) called back-translation. In the back-translation method, a synthetic corpus is generated by translating a large monolingual data in a target language into source sentences. For further evaluation and utilization of the extracted *Wikipedia* corpus, a comparison and adaptation the back-translation method is needed in future work.

SMT vs. NMT We compared the improvement of the *Wikipedia* corpus using the SMT versus NMT systems. Experimental results showed that the SMT systems obtained better performance on the *unconstrained* data (456k): 25.41 vs. 22.30 on the *tst2015*. Nevertheless, when the *Wikipedia* corpus was utilized, which was merged with the *unconstrained* data to enlarge the training data (864k), the NMT systems outperformed the SMT systems, which indicates the benefit when utilizing the *Wikipedia* corpus on NMT compared with SMT systems. Table 9 presents the comparison in more detail.

Model	**tst2012**	**tst2013**	**tst2015**
SMT systems			
unconstrained	34.42	27.19	25.41
unconstrained+Wikipedia	33.88	27.28 (+0.09)	26.36 (+0.95)
unconstrained*Wikipedia (tune)	34.44 (+0.02)	27.55 (+0.36)	26.68 (+1.27)
unconstrained*Wikipedia (weights)	34.73 (+0.31)	28.04 (+0.85)	26.78 (+1.37)
NMT systems			
unconstrained	24.05	26.71	22.30
unconstrained+Wikipedia	25.29 (+1.24)	28.93 (+2.21)	26.81 (+4.51)

Table 9: SMT versus NMT in using the *Wikipedia* corpus

From this comparison, we investigated the strategies to utilize the *Wikipedia* corpus most effectively for improving machine translation on low-resource languages, in which the corpus was utilized more effectively when using the NMT models.

7 Conclusion

Current machine translation systems in both phrase-based and neural-based methods require large bilingual corpora for training data. Nevertheless, such large bilingual corpora are unavailable for most language pairs called low-resource languages. This causes a bottleneck for the languages. In Southeast Asian languages, although there are a high population with more than five hundred millions of people, and there are several languages that can be used popularly in the world like Indonesian, Malay, and Vietnamese, there are few bilingual corpora on these language pairs, which causes a bottleneck for machine translation. In this paper, we introduce building a multilingual parallel corpus for several Southeast Asian languages of Indonesian, Malay, Filipino, Vietnamese, and the languages paired with English to improve machine translation. The corpus was built based on the Wikipedia's parallel titles of articles extracted by the articles' titles and inter-language link records. The parallel titles were used to collect parallel articles. For each article pair, parallel sentences were extracted based on a length-based and word correspondence sentence alignment method. A huge multilingual parallel corpus were obtained with more than 1.1 million parallel sentences of ten language pairs of the Southeast Asian languages. Experiments were conducted on the Asian Language Treebank and showed the promising results. Additionally, the corpus was utilized for the IWSLT 2015 machine translation shared task. A significant improvement was achieved on both phrase-based and neural-based systems with +1.7 and 4.5 BLEU points. The corpus can improve machine translation for the low-resource Southeast Asian languages and contribute to the development of machine translation on the low-resource languages.

Acknowledgement

This work was supported by JSPS KAKENHI Grant number JP15K16048 and the VNU project "Exploiting Very Large Monolingual Corpora for Statistical Machine Translation" (code QG.12.49).

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44. Association for Computational Linguistics.

Brown, P. F., Lai, J. C., and Mercer, R. L. (1991). Aligning sentences in parallel corpora. In *Proceedings of ACL*, pages 169–176. Association for Computational Linguistics.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Cettolo, M., Girardi, C., and Federico, M. (2012). Wit3: Web inventory of transcribed and translated talks. In Cettolo, M., Federico, M., Specia, L., and Way, A., editors, *Proceedings of th 16th International Conference of the European Association for Machine Translation (EAMT)*, pages 261–268.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Cattoni, R., and Federico, M. (2015). The iwslt 2015 evaluation campaign. *Proc. of IWSLT, Da Nang, Vietnam*.

Chen, S. F. (1993). Aligning sentences in bilingual corpora using lexical information. In *Proceedings of ACL*, pages 9–16. Association for Computational Linguistics.

Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proc. of HLT/NAACL*, pages 427–436. Association for Computational Linguistics.

Chu, C., Nakazawa, T., and Kurohashi, S. (2015). Integrated parallel sentence and fragment extraction from comparable corpora: A case study on chinese–japanese wikipedia. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 15(2):10:1–10:22.

Gale, W. A. and Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proc. of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Irvine, A. (2013). Statistical machine translation in low resource settings. In *Proceedings of HLT/NAACL*, pages 54–61. Association for Computational Linguistics.

Kay, M. and Röscheisen, M. (1993). Text-translation alignment. *Computational Linguistics*, 19(1):121–142.

Kim, S., Toutanova, K., and Yu, H. (2012). Multilingual named entity recognition using parallel data and metadata from wikipedia. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 694–702. Association for Computational Linguistics.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, Phuket, Thailand.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL*, pages 177–180. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Li, B. and Liu, J. (2008). Mining Chinese-English parallel corpora from the web. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP)*.

Ma, X. (2006). Champollion: A robust parallel text sentence aligner. In *Proceedings of LREC*, pages 489–492.

Melamed, I. D. (1996). A geometric approach to mapping bitext correspondence. In *Proceedings EMNLP*. Association for Computational Linguistics.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Munteanu, D. S. and Marcu, D. (2006). Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 81–88. Association for Computational Linguistics.

Ngo, Q. H., Winiwarter, W., and Wloka, B. (2013). Evbcorpus-a multi-layer english-vietnamese bilingual corpus for studying tasks in comparative linguistics. In *Proceedings of the 11th Workshop on Asian Language Resources (11th ALR within the IJCNLP2013)*, pages 1–9.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318. Association for Computational Linguistics.

Resnik, P. (1999). Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Sennrich, R. (2012). Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of EAMT*, pages 539–549.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation (WMT)*.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Singh, A. K. and Husain, S. (2005). Comparison, selection and use of sentence alignment algorithms for new language pairs. In *Proceedings of the ACL Workshop on Building and using Parallel texts*, pages 99–106. Association for Computational Linguistics.

Ştefănescu, D. and Ion, R. (2013). Parallel-wiki: A collection of parallel sentences extracted from wikipedia. In *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2013)*, pages 24–30.

Thu, Y. K., Pa, W. P., Utiyama, M., Finch, A., and Sumita, E. (2016). Introducing the asian language treebank (alt). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1574–1578.

Utiyama, M. and Isahara, H. (2003). Reliable measures for aligning Japanese-English news articles and sentences. In Hinrichs, E. and Roth, D., editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 72–79.

Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L., and Trón, V. (2007). Parallel corpora for medium density languages. *Amsterdam studies in the theory and history of linguistic science series 4*, 292:247.

Wang, P., Nakov, P., and Ng, H. T. (2016). Source language adaptation approaches for resource-poor machine translation. *Computational Linguistics*.

Weber, G. (2008). Top languages. *The World's*, 10.

Wu, D. (1994). Aligning a parallel english-chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 80–87. Association for Computational Linguistics.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *CoRR*.

Exploring Hypotheses Spaces in Neural Machine Translation

Frédéric Blain f.blain@sheffield.ac.uk
Lucia Specia l.specia@sheffield.ac.uk
Pranava Madhyastha p.madhyastha@sheffield.ac.uk
Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, UK

Abstract

Both statistical (SMT) and neural (NMT) approaches to machine translation (MT) explore large search spaces to produce and score translations. It is however well known that often the top hypothesis as scored by such approaches may not be the best overall translation among those that can be produced. Previous work on SMT has extensively explored re-ranking strategies in attempts to find the best possible translation. In this paper, we focus on NMT and provide an in-depth investigation to explore the influence of beam sizes on information content and translation quality. We gather new insights using oracle experiments on the efficacy of exploiting larger beams and propose a simple, yet novel consensus-based, n-best re-ranking approach that makes use of different automatic evaluation metrics to measure consensus in n-best lists. Our results reveal that NMT is able to cover more of the information content of the references compared to SMT and that this leads to better re-ranked translations (according to human evaluation). We further show that the MT evaluation metric used for the consensus-based re-ranking plays a major role, with character-based metrics performing better than BLEU.

1 Introduction

There has a been a recent surge of interest and work in the field of end-to-end, encoder-decoder neural machine translation (NMT). In the last two years, such approaches surpassed the state-of-the-art results by the then *de facto* statistical machine translation approaches (SMT) (Bojar et al., 2016a). While NMT systems are trained end-to-end using as a single model, SMT systems use a pipeline-based approach that make use of several components. This means that NMT systems are jointly optimised for both better encoding and better decoding. SMT systems, on the other hand, decompose the problem by first finding plausible sub-sentence translation candidates given some training data, such as phrases in phrase-based SMT (Koehn et al., 2003), and then scoring such candidates utilising components such as the translation and language models. Both types of systems are markedly different in their approaches to transform source into target language and in the information they explore.

Given a source sentence, at decoding time both types of approaches can explore hypotheses spaces to pick the best possible translation. Most of current implementations of both statistical and neural MT approaches use beam search for that. It has been observed that NMT systems, when compared to their statistical counterparts, use smaller beam sizes, and yet are able to obtain better translations for the same source sentences (Bahdanau et al., 2014; Stahlberg et al., 2017). Smaller beam sizes boost the speed of decoders (Luong et al., 2015; Bahdanau et al., 2014). In addition, it has been reported (Stahlberg et al., 2016) that neural approaches do not

significantly benefit from large beam sizes. In fact, beam sizes of 8–12 are the most common in NMT. Statistical approaches, on the other hand, usually search over larger beam sizes (of orders of 100s) (Lopez, 2008).

There have been multiple approaches proposed in the context of SMT that explore the n-best generated translation hypotheses using beam search (Och et al., 2004; Shen et al., 2004; Lambert and Banchs, 2006; Hasan et al., 2007; Duh and Kirchhoff, 2008). Since models used for scoring translation hypotheses and metrics used to evaluate the final translation quality are different, one of the strategies is to learn a re-ranking model for n-best hypotheses based on the evaluation metric of interest. We further detail this and other strategies in Section 2. However, to the best of our knowledge, there is little research that systematically looks at the effect of beam sizes or explores n-best hypotheses in the context of NMT.

We summarise our contributions in this paper as follows: (a) We investigate the influence of beam size on the search space, as well as on the information content of translations (Section 4); and (b) We present a new re-scoring approach for n-best re-ranking based on information overlap amongst MT candidates within the n-best list according to different automatic MT evaluation metrics. We report results that include human evaluation to assess the quality of alternative translations produced by this approach versus baseline systems (Section 5). We observe that our approach leads to better translation choices. We also observe that in most cases the best translation hypothesis is chosen among those generated from using larger beam sizes. These results are based on four language pairs and different datasets and evaluation metrics (Section 3).

2 Background

In what follows, we briefly describe background on the decoding process in SMT and NMT approaches, as well as related work on exploring n-best lists for improved translation quality.

Beam search decoding in SMT In SMT decoding, the standard procedure is to perform the search for the best translation given the (often pruned) space of possible translations based on a combination of the scores estimated for its model components, each component capturing a different aspect of translation (word order, translation probability, etc.). This is done through a heuristic method using stack-based beam search. In phrase-based SMT (Koehn et al., 2003), given a source sentence, the decoder fetches phrase translations available in the phrase table and builds a graph starting with an initial state where no source words have been translated and no target words have been generated. New states are created in the graph by extending the target output with a phrase translation that covers some of the source words not yet translated. At every expansion, the current cost of the new state is the cost of the original state multiplied with the model components under consideration. Final states in the search graph are hypotheses that cover all source words. Among these, the hypothesis with the lowest cost (highest model score) is selected as the best translation. Often a threshold is used to define a *beam* of good hypotheses and prune the hypotheses that fall out of this beam. The beam follows the (presumably) best hypothesis path, but with a certain width to allow the retention of comparable hypotheses, i.e. neighbouring hypotheses that are close in score from the best one (Koehn, 2010).

If an exhaustive search was to be performed, then all translation options, in different orders, could be used to build alternative hypotheses. However, in practice the search space is pruned in different ways and only the most promising hypotheses are kept, with early pruning potentially eliminating good hypotheses from the search space. In principle, larger beams would thus allow for more variation in the n-best lists, while potentially introducing lower quality candidates, but also giving seemingly *bad* candidates a chance to obtain higher scores in later stages of decoding. There is therefore a direct relationship between the size of the beam and the maximum number of candidates that can be generated in the n-best list. However, the actual

candidates in the n-best list are also affected by other design choices, such as the pruning and hypotheses combination strategies used (Lambert and Banchs, 2006; Duh and Kirchhoff, 2008; Hasan et al., 2007).

In addition, different approaches have been proposed to specifically promote diverse translations in SMT systems' n-best lists. These include using compact representations like lattices and hypergraphs (Tromble et al., 2008; Kumar and Byrne, 2004) and establishing explicit conditions during decoding. Gimpel et al. (2013), for example, add a dissimilarity function based on n-gram overlaps, choosing translations that have high model scores but are distinct from already-generated ones.

Beam search decoding in NMT NMT decoding also relies on beam search, but the process is much more expensive than in SMT and thus a limited beam size is often used, leading to narrow hypotheses spaces (Li and Jurafsky, 2016; Vijayakumar et al., 2016). Given a certain pre-specified beam size k, k-best lists are generated in a greedy left-right fashion retaining only the top-k candidates as follows: at the first time step in decoding, a fixed-number k hypotheses are retained based on the highest log-probability (model score) of each generated word. Each of the k hypotheses is expanded at each time-step by selecting top k word translations. This continues until the end-of-sequence symbol is obtained. The highest scoring candidate is retained and stored into the final candidate list followed by a decrease of beam by one. The whole process continues until the beam is reduced to zero. Finally, the best translation hypothesis amongst the list is the one with highest log-probability. We note here that in most NMT approaches both the set of hypotheses and the beam size are equivalent. Essentially, the NMT decoder obtains the top translation hypotheses that maximise the conditional probability given by the model.

Li and Jurafsky (2016) increase diversity in the n-best list by adding an additional component to the score used by the decoder to rank k hypotheses at each time step. This component rewards top-ranked hypotheses generated from each ancestor, instead of ranking all candidates from all ancestors together. Similarly, Vijayakumar et al. (2016) propose *Diverse Beam Search*, where they optimise an objective with two terms: the standard cross entropy loss and a dissimilarity term that encourages beams across groups to differ.

N-best re-ranking in SMT In addition to having access to only a subset of the search space, the model components used in SMT only provide an estimate of translation quality. As a consequence, using only the hypothesis ranked as the best by the decoder often leads to suboptimal results (Wisniewski et al., 2010; Sokolov et al., 2012a). Therefore, it is common practice in SMT to explore other hypotheses in the search space, the so called *n-best list*. Re-ranking an n-best list of candidates produced by an SMT system has been a long standing practice. The general motivation for doing so is the ability to use additional information in the process, which is unavailable or too costly to compute at decoding time, e.g. syntactic features of the entire sentence (Och et al., 2004), estimates on overall sentence translation quality (Blatz et al., 2003), word sense disambiguation scores (Specia et al., 2008), large language model scores (Zhang et al., 2006), and translation probability from a neural MT model (Neubig et al., 2015), among others.

This additional information is usually treated as new model components and combined with the existing ones. Various techniques have been proposed to perform n-best list re-ranking. They generally learn weights to combine the new and existing model components using algorithms such as MIRA (Crammer and Singer, 2003) with linear[1] or non-linear functions (Sokolov et al., 2012b), as well as more advanced methods, such as multi-task learning (Duh et al., 2010). Hasan et al. (2007) provides a study on the potential improvements on final translation quality by exploring n-best lists of different sizes. They show that even though oracle-based re-ranking

[1] `https://github.com/moses-smt/mosesdecoder/tree/master/scripts/nbest-rescore`

on very large (100,000 hypotheses) n-best lists yields the best translation quality, automatic re-ranking methods reach a plateau on the improvement after 1,000 hypotheses. Very large n-best lists will contain very many noisy translations, so they suggest that only with extremely accurate re-ranking methods one should explore such large spaces.

In an attempt to have a more reliable way to score translation candidates, Kumar and Byrne (2004) introduced the Minimum Bayes Risk (MBR) decoding approach and used it to re-rank n-best hypotheses such that the best hypothesis is the one that minimises the Bayes-risk defined in terms of the model score (translation probability) and a loss function computed between the translation hypothesis and a gold translation (e.g. a translation quality metric such as BLEU (Papineni et al., 2002)). This method has been shown to be beneficial for many translation tasks (Ehling et al., 2007; Tromble et al., 2008; Blackwood et al., 2010). They have however only experimented a fixed n (1,000).

N-best re-ranking in NMT While there is a large body of literature that investigates different strategies for exploring n-best hypotheses spaces in SMT, there have been very few attempts at exploring such spaces in NMT. Stahlberg et al. (2017) adapt MBR decoding to the context of NMT and to be used for partial hypotheses rather than entire translations. The NMT score is combined with the Bayes-risk of the translation according to the SMT lattice. This approach goes beyond re-scoring of n-best lists or lattices as the neural decoder is not restricted to the SMT search space. The resulting MBR decoder produces new hypotheses that are different from those in the SMT search space.

Li and Jurafsky (2016) propose an alternative objective function for NMT that maximises the mutual information between the source and target sentences. They implement the model with a simple re-ranking method. This is equivalent to linearly combining the probability of the target given the source, and vice-versa. An NMT model is trained for each translation direction, and the source→target model is used to generate n-best lists. These are then re-ranked using the score from the target→source model. Shu and Nakayama (2017) studies the effect of beam size in NMT MBR decoding. They considered beams of size 5, 20 and 100 and found that while in standard decoding increasing the beam size is not beneficial, MBR re-ranking is more effective with a large beam size.

Comparison between NMT and SMT There has been increasing interest in systematically studying differences between NMT and SMT approaches. Bentivogli et al. (2016) conducted an analysis for English→German translations by both NMT and SMT systems. They conclude that the outputs of the NMT system are better suited in terms of syntax and semantics, with better word order and less human post-editing effort required to fix the translations. They observe that the average sentence length in an SMT system is always longer than in an NMT system. This could be attributed to the optimisation of the cross-entropy loss and the fact that the outputs are chosen on the basis of the log-probability scores in NMT systems.

Toral and Sánchez-Cartagena (2017) conducted an in-depth analysis on a set of nine language pairs to contrast the differences between SMT and NMT systems. They observe that the outputs of NMT systems are more fluent and have better word order when compared to SMT systems. They note that despite the smaller beam sizes in NMT in general the top outputs of the NMT system for a given source sentence are more distinct than the top outputs from SMT systems. However, it is not clear whether or not they explore distinct n-best options from the SMT or a mixture of distinct and non-distinct options. Both previous studies conclude that the NMT systems perform poorly when translating very long sentences.

3 Experimental Settings

In this section we describe the data, tools, metrics and settings used in our experiments to investigate the influence of beam size in the generated translations.

Language Pairs We report results with NMT systems – the focus of this paper – for four language pairs: English↔German and English↔Czech. For English↔Czech we also report results with SMT systems for comparison.

NMT Systems We use the freely available Nematus (Sennrich et al., 2016) toolkit and its pre-trained models[2] for English↔German and English↔Czech. The Nematus systems are based on attentional encoder-decoder neural machine translation approach (Bahdanau et al., 2014) and were built after *Byte-Pair Encoding* (Sennrich et al., 2015b).[3] The models were trained as described in (Sennrich et al., 2016) using both parallel and synthetic (Sennrich et al., 2015a) data under the constrained variant of the WMT16 MT shared task, mini batches of size 80, a maximum sentence length of 50, word-embeddings of size 500, a hidden layers of size 1024, and Adadelta as optimiser (Zeiler, 2012), reshuffling the training corpus between epochs.These models were chosen as they have been highly ranked in the evaluation campaign of the WMT16 Conference (Bojar et al., 2016c).

SMT Systems We use pre-trained models from the Tuning shared task of WMT16 for English↔Czech to build SMT systems for comparison. These models were built using the Moses toolkit (Koehn et al., 2007) trained on the CzEng1.6pre[4], (Bojar et al., 2016b) a 51M parallel sentences corpus built from eight different sources. The data was tokenised using Moses tokeniser (Koehn et al., 2007) and lowercased; sentences longer than 60 words and shorter than 4 words were removed before training. The weights were determined as the average over three optimisation runs using MIRA (Crammer and Singer, 2003) towards BLEU. Word alignment was done using fast-align (Dyer et al., 2013) and for all other steps the standard Moses pipeline was used for model building and decoding. This was reported as the best system for English↔Czech (Jawaid et al., 2016).

By using pre-trained and freely available models for our NMT and SMT systems, we have consistent models amongst the different language pairs and results can be more easily reproducible.

Beam Settings SMT systems usually employ a large beam. In the training pipeline of the Moses decoder, the beam size is set by default to 200. NMT systems, on the other hand, normally use a much smaller beam size of 8 to 12. This is assumed to offer a good trade off between quality and computational complexity. We note that the implementations of n-best decoding is different in both NMT and SMT. In most NMT systems, there is a 1-to-1 correspondence between the beam size and the n-best list size. Therefore, we will use the term n-*best* to refer to the output of an NMT system with a beam of size n, and to the n best outputs of an SMT system, where the beam size has been set, by default, to 200.

We also note that the translations in the n-best list produced by NMT are always different from each other, even though only marginally in many cases (e.g. a single token). In SMT, one can choose whether or not only distinct candidates should be considered. We report on distinct options only to gather insights on the diversity in n-best lists in SMT versus NMT.

Metrics For our experiments we consider three automatic evaluation metrics amongst the most widely used and which have been shown to correlate well with human judgements (Bojar

[2] `http://data.statmt.org/rsennrich/wmt16_systems/`

[3] The models were obtained from `http://statmt.org/rsennrich/wmt16_systems/`

[4] `http://ufal.mff.cuni.cz/czeng/czeng16pre`

et al., 2016c): **BLEU**, an n-gram-based precision metric which works similarly to position-independent word error rate, but considers matches of larger n-grams with the reference translation; **BEER** (Stanojevic and Sima'an, 2014), a trained evaluation metric with a linear model that combines features capturing character n-grams and permutation trees; and **ChrF** (Popovic, 2015), which computes the F-score of character n-grams. These metrics are used both for evaluating final translation quality and for measuring similarity among translations in our consensus-based re-ranking approach.

4 Effect of Beam Size

Current work in NMT takes a beam size of around 10 to be the optimal setting (Sennrich et al., 2016). We empirically evaluate the effect of increasing the beam size in NMT to explore n-best of sizes 10, 100 and 500. The goals are to understand (a) the informativeness of the translations produced; (b) the scope for obtaining better translations by simply exploiting the n-best candidates, similarly to previous work in SMT.

4.1 Effect of Beam Size on Information Content of Translations

We define information content as the word overlap rate between the system generated translation and the reference translation. We further break this into two categories:

1. *% covered*: This indicates the average proportion of words that are shared between the (a) 1-best output of the MT system and the reference translation, or (b) all the n-best outputs and the reference translation. It is computed by looking at the intersection between the vocabulary of the MT candidate(s) and the one of the reference, averaged at corpus-level.

2. *% exact match*: This indicates the proportion of sentences that are exact matches between (a) the 1-best of the MT system and the reference translation, and (b) all the n-best outputs and the reference translation.

This is similar to the approach in (Lala et al., 2017) where the authors measure word overlap with respect to system outputs, but their focus is on multimodal NMT. *% covered* approximates indicates the word-level precision of the MT system, given the n or 1-best candidates and the reference translation, and *% exact match* approximately indicates the sentence-level recall given the n or 1-best candidates and the reference translation.

Our intuition here is that if the systems are adequately trained, increasing the beam size – and thereby the n-best list length – should result in obtaining a larger word overlap with reference translation, and potentially a larger number of exact matches at the sentence level, although the latter is a much taller order. We note that since only one reference translation is available, mismatches between words in the MT output and reference translations could reflect acceptable variances in translation.

Observations and Discussion In Table 1 we report the scores of each MT system using BLEU, BEER and ChrF3 on the WMT16 test sets with different sizes of n-best lists: for NMT we report sizes 10, 100 and 500, while for SMT we report a 500-best list with a beam size set to the default size of 200. Since there is no 1-to-1 relationship between beam sizes and n-best list sizes in SMT, reporting on different beam sizes would require arbitrarily choosing a specific n for each beam size. We instead chose the largest n also used for the NMT experiments (500), and a large enough beam size (200). The metric scores are computed on the 1-best translation, which may vary if different beam sizes are used. We observe that for NMT increasing the n-best size from 10 to 100 helps improve the performances for English↔German translations. For English↔Czech, we do not observe any gain, but rather a significant drop. Also, if the beam size is too large (500 in our case), the performance drops for all language pairs. This indicates

that larger beam sizes do not necessarily lead to better 1-best translations, and that the choice can be a function of the language pair and the dataset. This seems to suggest that with such large beam sizes many translation candidates, including spurious ones, end up being ranked as the 1-best, most likely because of limitations in the functions used to score translation candidates.

NEURAL MT	**English→German**			**German→English**		
n-best	BLEU	BEER	ChrF3	BLEU	BEER	ChrF3
n=10	26.73	60.20	59.20	32.58	61.84	60.61
n=100	26.82	60.25	59.33	32.68	61.91	60.74
n=500	26.18	60.12	59.12	32.70	61.91	60.75
	English→Czech			**Czech→English**		
n-best	BLEU	BEER	ChrF3	BLEU	BEER	ChrF3
n=10	18.50	53.90	51.45	26.26	58.03	56.00
n=100	18.31	53.83	51.37	26.17	58.00	56.00
n=500	17.81	53.67	51.25	24.19	57.57	55.62

STATISTICAL MT	**English→Czech**			**Czech→English**		
n-best	BLEU	BEER	ChrF3	BLEU	BEER	ChrF3
n=10/100/500	10.64	48.88	46.51	18.19	52.59	51.32

Table 1: Translation quality results on the WMT16 test sets for both NMT and SMT systems using n-best lists of sizes 10, 100 and 500. The scores are computed on the 1-best translation towards the reference translation.

In Table 2 we report our empirical observations on word coverage. Here, we observe that the larger the n-best list the higher proportion of words covered (*% covered*). Interestingly, we also observe similar trends for *% exact match*, but only if all n-best candidates are considered. It also interesting to note the difference in the impressive increase in *% exact match* from 1-best to all-best for NMT, which does not happen for SMT. These results show that for NMT larger beam sizes lead to more information content in translation candidates. Therefore, clever techniques to explore the space of hypotheses should lead to better translations.

Even though the NMT vs SMT figures are not directly comparable since the NMT and SMT systems are trained on different data, we note that despite the SMT system using a beam size of 200 and producing 500-best translation hypotheses, its translations have much lower word overlap than those from the NMT system with a beam size of 10 for English↔Czech. These results further corroborate the reasons for the insignificant gains obtained in the WMT16 SMT system Tuning shared task (Jawaid et al., 2016). In fact, if larger hypotheses spaces do not lead to more words that can potentially lead to translations that match the reference, the tuning algorithms do not have much to learn from.

4.2 Oracle Exploration

Based on the encouraging observations in the previous experiment with word overlap between candidates in the n-best list and the reference translation, here we attempt to quantify the potential gain from optimally exploring the space of hypotheses. We perform experiments assuming that we have an 'oracle' which helps us choose the best possible translation, under an evaluation metric against the reference, given an n-best list of translation hypotheses. This provides an upper-bound on the performance of the MT system. Positive results in this experiment will indicate that the MT system is capable of producing better translation candidates, but fails at scoring them as the best ones.

In this oracle experiment, the translation of a source sentence is chosen based on comparisons among the translation hypotheses and the reference translation – the oracle – under a

NEURAL MT	10-best		100-best		500-best	
	1-best	*all*	*1-best*	*all*	*1-best*	*all*
English→German						
%covered	53.99	62.75	53.99	71.93	53.83	77.69
% exact match	2.20	6.47	2.20	12.07	2.20	18.24
German→English						
%covered	57.32	65.98	57.43	74.42	57.43	79.55
% exact match	2.70	7.70	2.70	15.40	2.70	22.94
English→Czech						
%covered	45.97	55.27	45.85	65.61	45.72	72.55
% exact match	1.63	4.90	1.63	9.40	1.63	14.77
Czech→English						
%covered	52.30	61.26	52.33	70.24	51.92	75.61
% exact match	1.67	14.44	1.67	11.47	1.60	16.97

STATISTICAL MT	10-best		100-best		500-best	
(beam=200, distinct)	*1-best*	*all*	*1-best*	*all*	*1-best*	*all*
English→Czech						
% covered	39.20	46.58	39.20	54.05	39.20	57.86
% exact match	0.07	0.07	0.07	0.37	0.07	0.37
Czech→English						
% covered	48.35	54.79	48.35	60.30	48.35	62.89
% exact match	0.16	0.50	0.16	0.83	0.16	0.83

Table 2: Proportion of words overlapping between candidates and reference translations for different values of the n-best, as well as proportion of MT output sentences that exactly match the reference, considering either the 1-best or all the MT candidates in the n-best list.

certain MT evaluation metric. We consider the outputs of NMT systems for beam sizes of 10, 100 and 500 and with the following metrics: BLEU with n-gram max length $= 4$ and default brevity penalty settings, BEER2.0 with default settings, and ChrF with n-gram max length $= 6$ and $\beta = 3$. By exploring multiple metrics we will gain insights on how well different metrics do at spotting the best candidates: ideally, better metrics should lead to larger improvements from the original top translation.

Observations and Discussion We report the results of the oracle experiment in Figure 1. For each system, we report the relative improvement (delta) between the oracle translation chosen by the three metrics – BLEU, BEER and ChrF3 – compared to the 1-best of the system for a given n-best list size. Using any of the metrics we are able to find an alternative MT candidate which is better than the original 1-best translation, resulting in an overall increase in translation quality in all datasets. Larger improvements are obtained with larger beam sizes. However, while a large gain (almost double) is obtained from beam size 10 to 100, the rate of increase in improvement seems to drop from beam size 100 to 500, indicating that more additional translations are probably mostly spurious. This is consistent with the information content experiment in Section 4.1.

Kumar and Byrne (2004) reports that their MBR decoder leads to improvements only according to an evaluation metric that is also used as basis for their loss function. In our experiments, to better understand the relationship between the re-ranking metric and the final evaluation results, we further explore the oracle experiment by reporting results on the 500-best output for NMT, which brings the best gains in Figure 1, but focus on the proportion of improvement of the oracle translation over 1-best *across metrics*. In other words, we oracle rerank using each given metric and evaluate the final 1-best translation set performance using all

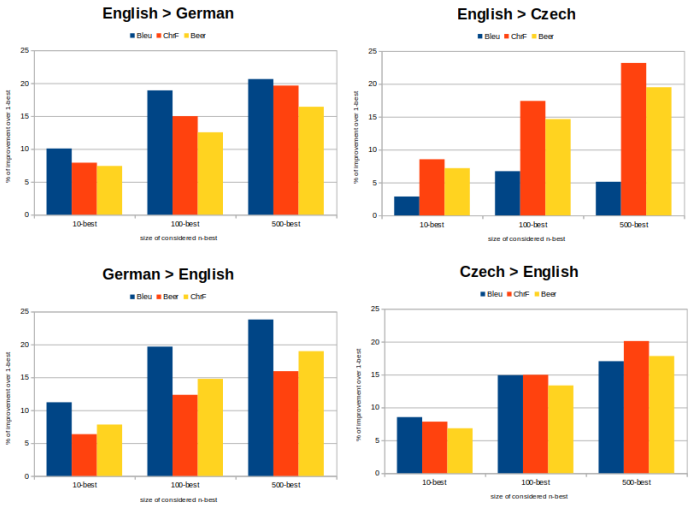


Figure 1: Proportion of improvement in NMT results according to MT evaluation metrics based on the oracle results over the original 1-best when the size of the beam is increased for decoding.

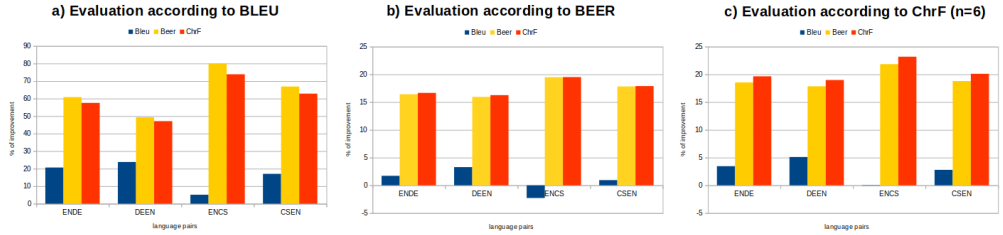


Figure 2: Focusing on the 500-best output for NMT, which brings the best gains in Figure 1, proportion of improvement of the oracle translation over the original 1-best when using different metrics for the oracle computation: ChrF3, BEER and BLEU. Re-ranking is done with one metric at a time, and the final performance is also measured with each of three metrics.

three metrics. This helps us assess the potential of each metric in selecting the best candidate. Figure 2 shows the results. Contrary to what was suggested in Kumar and Byrne (2004) for SMT, in chart (a) we see that the relative improvement is bigger in terms of the BLEU metric when using either BEER or ChrF3 to obtain the 1-best translation than using BLEU itself. We also observe in charts (b) and (c) that the character-based metrics always outperform BLEU and extract better 1-best translations. BLEU also seems to fail at identifying better MT candidates when translating into Czech, which is a morphologically rich language, while BEER and ChrF3 perform better. We note however that Kumar and Byrne (2004) also tune the log-linear loss function, while in our case we are just selecting the candidates directly based on a metric.

Since sentence length is a often problem in NMT, we measure the impact of using different evaluation metrics for oracle re-ranking on the sentence length of the 1-best translations chosen. In Figure 3 we report variation in terms of sentence length average for all NMT systems after the oracle translation selection with all three metrics, compared to the original 1-best translation for each setting. We notice that the average length of oracle BLEU translations does not seem to vary, however, an opposite trend is seen with BEER and ChrF3, which seem to make sentences

shorter except for German→English. This is particularly interesting since i) we observe in Table 2 a better coverage with bigger beam size, and ii) we observe an overall large BLEU improvement our oracle experiments (Figure 2 (a)). This suggests that we are able to select translation candidates that might be shorter than the original 1-best, but most similar to the reference translation.

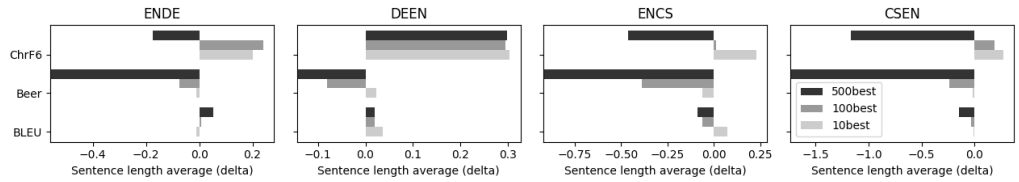


Figure 3: Delta in average sentence length for all NMT systems after 1-best oracle translation selection by each metric, compared to the average sentence length of the original 1-best.

5 Consensus-based n-best re-ranking

As was shown in the previous section, increasing the size of the beam generally leads to better word coverage and, more important, to higher chances of generating better translations among the resulting n-best lists. In what follows we propose an approach to automatically re-rank n-best lists to obtain better translations (without oracle translations).

Our approach is motivated by the work of DeNero et al. (2009) for SMT, where consensus-based MBR decoding is used to guide the choices of the decoder towards hypotheses that share partial translations. DeNero et al. (2009) experiment with different evaluation metrics (including BLEU) to measure similarity among hypotheses within a n-best list. We propose to empirically evaluate the contribution of consensus information in hypotheses in n-best lists from NMT systems. This is simpler than using consensual information at decoding time, but we believe that positive results at re-ranking stage will provide insights on whether or not this is a promising path to follow in NMT decoding.

Given an n-best list and a certain similarity metric, we compute the metric scores for each translation hypothesis against each of all $n-1$ other hypotheses in the n-best list. We then average the similarity scores of all $n-1$ translation hypotheses to obtain a single score for each translation hypothesis. We repeat this for all translation hypotheses and then sort the n-best list based on these scores, such that the top (best) translation will be one that is similar to more of the alternative candidates. Given that NMT systems produce translations are are "more likely" given the model, this essentially corresponds to selecting as best translation the one that is the most similar to all of $n-1$ the most likely translations. The size of the n-best list here is critical: the more hypotheses in the list, the less confident the NMT system will be on the bottom part of the list (less likely translations). However, longer n-best lists may provide stronger evidence for consensual analysis. This is a classical exploration-exploitation issue.

Another remark is that larger search spaces require much more time to compute the consensus-based re-ranking. We experiment with BLEU, BEER and ChrF3 as similarity metrics, since these are easily available and are either extremely popular (BLEU) or have proved to correlate well with human judgements on translation quality (in terms of similarity with a reference translation) in recent evaluation campaigns (BEER and ChrF3) (Bojar et al., 2016c). While each pair of translation hypotheses can be scored independently, which allows parallel processing, the running time for each metric to re-rank a complete n-best list is $O(n^2 \cdot k)$, where k is the size of the corpus and n the size of the n-best list. This may be very time consuming:

from hours up to a day[5] for easy-to-compute metrics such as BLEU or ChrF, to many days for more complex metrics such as BEER.

Automatic evaluation We start by evaluating our consensus-based re-ranking approach using BLEU as automatic evaluation metric. The results are shown in Table 3. A similar trend was observed using BEER and ChrF3 as similarity metrics, however we omit these results due to space constraints. Comparing the figures in this table against those in Table 1, we see that – under the same beam size – re-ranking seems to degrade the results in all cases with BLEU and ChrF, but not with BEER. An increase in BLEU scores can be observed for BEER-based re-ranking as longer beam sizes superior to 10 are used for the two language pairs where re-ranking under this metric was computed. It is not surprising to see that this improvement is only observed for BEER as similarity metric, even though the final evaluation is in terms of BLEU. This suggests that exploring other similarity metrics for the consensus analysis could be beneficial. Overall, re-ranking using BEER as similarity metric leads to the best results.

	English→German				**German→English**			
		re-ranked with				*re-ranked with*		
n-best	*baseline*	BLEU	BEER	ChrF3	*baseline*	BLEU	BEER	ChrF3
n=10	26.93	26.51	26.77	26.38	32.58	32.10	32.29	31.79
n=100	26.82	26.02	26.87	26.18	32.68	31.90	32.78	31.67
n=500	26.18	24.80	-	25.93	32.70	31.41	32.85	32.25
	English→Czech				**Czech→English**			
		re-ranked with				*re-ranked with*		
n-best	*baseline*	BLEU	BEER	ChrF3	*baseline*	BLEU	BEER	ChrF3
n=10	18.50	17.98	18.24	17.60	26.26	25.81	26.10	25.52
n=100	18.31	17.58	18.61	17.57	26.17	25.47	26.42	25.16
n=500	17.81	16.39	-	17.38	24.19	24.44	26.57	24.80

Table 3: BLEU scores of our consensus-based re-ranking strategy on the WMT16 test sets with NMT using n-best lists of sizes 10, 100 and 500. The scores are computed on the newly ranked 1-best NMT candidate against the reference translation. The *baseline* scores correspond to the original 1-best assessed towards the reference translation (see Table 1). The current implementation of BEER makes our consensus-based re-ranking extremely time consuming and virtually unfeasible, therefore we only show results for a subset of language pairs.

In Table 4 we illustrate some examples from the re-ranking approach. We observed that the consensus-based re-ranking produced interesting sentences that included syntactic re-orderings, new words, morphological variations and other nuances which were not captured by BLEU. This motivated us to perform human evaluation of the translations to more quantitatively compare the original 1-best versus the re-ranked 1-best.

Human evaluation We conducted a human evaluation using Appraise (Federmann, 2012), an open-source web application for manual evaluation of MT output. Appraise collects human judgements on translation output, implementing annotation tasks such as quality checking, error classification, manual post-editing and, in our case, translation ranking. For a list of up to four systems' outputs for each source sentence, we requested human annotators to rank the set of MT candidates from the best to the worst, allowing for ties, based on both the source sentence and reference translation. If two system outputs are the same, the MT candidate was displayed once and the same rank was assigned to both systems.

For this evaluation, we selected a subset of our systems based on our automatic evaluation results: for each metric used for re-ranking in each language pair, we chose the systems that

[5]Indicative time it took to re-rank a corpus of 3,000 sentences, with $n = 500$ on a 40-cores CPU server.

	German→English
SRC:	Das rund zehn bis zwölf Millionen Euro teure Vorhaben steht seit Monaten in der Diskussion.
REF:	The € 10 - 12 million project has been under discussion for months.
Baseline:	the EUR 10 million project has been under discussion for months.
BEER:	the approximately EUR 10 to 12 million projects has been under discussion for months
ChrF3:	the EUR 10 million euro project has been under discussion for several months.
BLEU:	the projects around ten to twelve million euros have been discussed for months.
	Czech→English
SRC:	Navíc jsem si ze života odnesl zkušenost, že zasahování do ekosystému nevede k úspěchu a jednoho škůdce může nahradit druhý.
REF:	Furthermore, in my experience, interfering with the ecosystem does not lead to success and one pest can replace another.
Baseline:	moreover, I have learned from life that interfering with an ecosystem does not lead to success, and one pest can replace another.
BEER:	moreover, I have learned from my life that it is not possible to succeed in an ecosystem, and one can replace one of the pests.
ChrF3:	moreover, I have learned from life that interfering with an ecosystem does not lead to success, and one pest can replace one another.
BLEU:	moreover, I have learned from life that interfering with an ecosystem does not lead to success, and one pest can replace one.

Table 4: Examples of alternative MT candidates chosen by consensus from n-best lists (with $n = 500$). Boxes highlight the main differences between the reference translation, the baseline (i.e. the original 1-best) and an alternative translation chose by our consensus re-ranking approach using BLEU, BEER or ChrF.

performed the best according to the three metrics (averaged ranking among the three), along with the original 1-best.

Each human translator was asked to complete at least one hit of twenty annotation tasks. Incomplete hits were discarded from the evaluation. We collected 3,016 complete ranking results over the four NMT systems (159 for English→Czech, 1,365 for Czech→English, 911 for English→German, 581 for German→English), from 208 annotators.

We borrowed a method from the WMT translation shared task to generate a global ranking of systems from these judgements. Table 5 reports the ranking results according to the Expected Wins method[6] for the four language pairs. The first column ($\#_m$) indicates the ranking of the systems amongst themselves according to the three automatic metrics, while the third column (range) indicates the ranking from the human evaluation. For example, for English→German, the *BLEU-100best* system was ranked first amongst the four by all three metrics, but it was ranked last by human annotators.

[6] https://github.com/keisks/wmt-trueskill/blob/master/src/infer_EW.py

English→German

$\#_m$	score	range	system
4	0.578	1-2	BEER-100best
2	0.529	1-3	Baseline (10best)
3	0.505	2-3	ChrF3-10best
1	0.388	4	BLEU-100best

German→English

$\#_m$	score	range	system
4	0.559	1-3	BEER-500best
2	0.546	1-3	Baseline (10best)
3	0.525	1-3	ChrF3-10best
1	0.393	4	BLEU-500best

English→Czech

$\#_m$	score	range	system
2	0.583	1-3	BEER-100best
4	0.532	1-3	ChrF3-100best
1	0.493	1-4	BLEU-100best
3	0.372	3-4	Baseline (100best)

Czech→English

$\#_m$	score	range	system
4	0.526	1-3	BEER-10best
3	0.522	1-2	ChrF3-500best
2	0.508	1-3	Baseline (500best)
1	0.453	3-4	BLEU-500best

Table 5: Results of the human evaluation for NMT. Systems are sorted according to human assessments while $\#_m$ indicates the overall ranking of a system according to all three automatic metrics. Scores and ranges are obtained with the Expected Wins method (Sakaguchi et al., 2014). Lines between systems indicate clusters. Systems within a cluster are considered tied. In gray are systems which have not significantly outperformed the baseline.

Our first observation is that the consensus-based re-ranking with BEER outperforms the other two metrics for all the language pairs, confirming the results of the automatic evaluation. Except for Czech→English, systems always benefit from a beam size larger than 10, which suggests that we should consider exploiting a larger search spaces in NMT. Another interesting outcome of the human evaluation is the ranking of our systems, which for most of the language pairs refutes the ranking according to the automatic evaluation. Although those metrics are known to be well correlated with human judgements, it seems that humans have different perceptions on the quality of the translations.

6 Conclusions

In this paper we reported our experiments and results on the influence of the beam size in NMT. While traditional approaches in NMT rely on smaller beam sizes or use greedy implementations, our paper strongly motivates using a larger beam size. We investigate the informativeness of larger beam size and highlighted the potential to improve translation quality by exploring larger hypotheses spaces using an oracle experiment. Motivated by substantial potential gains in both informativeness and oracle-based hypotheses re-ranking, we proposed a consensus-based NMT n-best re-ranking approach, with insights into the use of different metrics to capture consensus-based information. Our contribution strongly suggests further work in NMT to explore larger beams and n-best lists.

Acknowledgements

This work was supported by the QT21 project (H2020 No. 645452).

References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bentivogli, L., Bisazza, A., Cettolo, M., and Federico, M. (2016). Neural versus phrase-based machine translation quality: a case study. *arXiv preprint arXiv:1608.04631*.

Blackwood, G., de Gispert, A., and Byrne, W. (2010). Efficient path counting transducers

for minimum bayes-risk decoding of statistical machine translation lattices. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 27–32, Uppsala, Sweden.

Blatz, J., Fitzgerald, E., Foster, G., Simona Gandrabur, C. G., Kulesza, A., Sanchis, A., and Ueffing, N. (2003). Confidence estimation for machine translation. Technical report, Johns Hopkins University.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Neveol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., and Zampieri, M. (2016a). Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany.

Bojar, O., Dušek, O., Kocmi, T., Libovický, J., Novák, M., Popel, M., Sudarikov, R., and Variš, D. (2016b). CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *19th International Text, Speech and Dialogue Conference*, Brno, Czech Republic. Springer Verlag.

Bojar, O., Graham, Y., Kamran, A., and Stanojevic, M. (2016c). Results of the wmt16 metrics shared task. In *Proceedings of the First Conference on Machine Translation*, Berlin, Germany.

Crammer, K. and Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991.

DeNero, J., Chiang, D., and Knight, K. (2009). Fast consensus decoding over translation forests. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Conference on Natural Language Processing of the AFNLP*, pages 567–575.

Duh, K. and Kirchhoff, K. (2008). Beyond log-linear models: boosted minimum error rate training for n-best re-ranking. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 37–40.

Duh, K., Sudoh, K., Tsukada, H., Isozaki, H., and Nagata, M. (2010). N-best reranking by multitask learning. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 375–383, Uppsala, Sweden. Association for Computational Linguistics.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of IBM model 2.

Ehling, N., Zens, R., and Ney, H. (2007). Minimum Bayes risk decoding for BLEU. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 101–104, Prague, Czech Republic.

Federmann, C. (2012). Appraise: An open-source toolkit for manual evaluation of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Gimpel, K., Batra, D., Dyer, C., and Shakhnarovich, G. (2013). A systematic exploration of diversity in machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Hasan, S., Zens, R., and Ney, H. (2007). Are very large n-best lists useful for smt? In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 57–60, Rochester, New York.

Jawaid, B., Kamran, A., Stanojević, M., and Bojar, O. (2016). Results of the wmt16 tuning shared task. In *Proceedings of the First Conference on Machine Translation*, pages 232–238, Berlin, Germany.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics: Demon Session*, pages 177–180.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, Canada.

Kumar, S. and Byrne, W. (2004). Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics*, pages 169–176, Boston, MA.

Lala, C., Madhyastha, P., Wang, J., and Specia, L. (2017). Unraveling the contribution of image captioning and neural machine translation for multimodal machine translation. *The Prague Bulletin of Mathematical Linguistics*.

Lambert, P. and Banchs, R. (2006). Tuning machine translation parameters with spsa. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 190–196.

Li, J. and Jurafsky, D. (2016). Mutual information and diverse decoding improve neural machine translation. *CoRR*, abs/1601.00372.

Lopez, A. (2008). Statistical machine translation. *ACM Computing Surveys*, 40(3):8:1–8:49.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Neubig, G., Morishita, M., and Nakamura, S. (2015). Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. *CoRR*, abs/1510.05203.

Och, F. J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A. M., Kumar, S., Shen, L., Smith, D., Eng, K., et al. (2004). A smorgasbord of features for statistical machine translation. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics*, pages 161–168.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Popovic, M. (2015). chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the 10th Workshop on Statistical Machine Translation*, pages 392–395.

Sakaguchi, K., Post, M., and Van Durme, B. (2014). Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 1–11.

Sennrich, R., Haddow, B., and Birch, A. (2015a). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Sennrich, R., Haddow, B., and Birch, A. (2015b). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany.

Shen, L., Sarkar, A., and Och, F. J. (2004). Discriminative reranking for machine translation. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics*, pages 177–184.

Shu, R. and Nakayama, H. (2017). Later-stage Minimum Bayes-Risk Decoding for Neural Machine Translation. *ArXiv e-prints*.

Sokolov, A., Wisniewski, G., and Yvon, F. (2012a). Computing lattice bleu oracle scores for machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 120–129, Avignon, France.

Sokolov, A., Wisniewski, G., and Yvon, F. (2012b). Non-linear n-best reranking with few features. In *Proceedings of the 10th Conference of the Association for Machine Translation in the Americas*, pages 1–10, San Diego, CA.

Specia, L., Sankaran, B., and Graças Volpe Nunes, M. (2008). n-best reranking for the efficient integration of word sense disambiguation and statistical machine translation. *Lecture Notes in Computer Science*, 4919:399–410.

Stahlberg, F., de Gispert, A., Hasler, E., and Byrne, B. (2017). Neural machine translation by minimising the bayes-risk with respect to syntactic translation lattices. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 362–368, Valencia, Spain.

Stahlberg, F., Hasler, E., Waite, A., and Byrne, B. (2016). Syntactically guided neural machine translation. *arXiv preprint arXiv:1605.04569*.

Stanojevic, M. and Sima'an, K. (2014). Beer: Better evaluation as ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419.

Toral, A. and Sánchez-Cartagena, V. M. (2017). A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. *arXiv preprint arXiv:1701.02901*.

Tromble, R. W., Kumar, S., Och, F., and Macherey, W. (2008). Lattice minimum bayes-risk decoding for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 620–629, Honolulu, Hawaii.

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D. J., and Batra, D. (2016). Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

Wisniewski, G., Allauzen, A., and Yvon, F. (2010). Assessing phrase-based translation models with oracle decoding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 933–943, Cambridge, Massachusetts.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhang, Y., Hildebrand, A. S., and Vogel, S. (2006). Distributed language modeling for n-best list re-ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 216–223, Sydney, Australia.

Confidence through Attention

Matīss Rikters matiss@lielakeda.lv
Faculty of Computing, University of Latvia

Mark Fishel fishel@ut.ee
Institute of Computer Science, University of Tartu, Estonia

Abstract

Attention distributions of the generated translations are a useful bi-product of attention-based recurrent neural network translation models and can be treated as soft alignments between the input and output tokens. In this work, we use attention distributions as a confidence metric for output translations. We present two strategies of using the attention distributions: filtering out bad translations from a large back-translated corpus, and selecting the best translation in a hybrid setup of two different translation systems. While manual evaluation indicated only a weak correlation between our confidence score and human judgments, the use-cases showed improvements of up to 2.22 BLEU points for filtering and 0.99 points for hybrid translation, tested on English↔German and English↔Latvian translation.

1 Introduction

Neural machine translation (NMT) has recently redefined the state-of-the-art in machine translation (Sennrich et al., 2016a; Wu et al., 2016a), with one of the ground-breaking innovations that enabled this being the introduction of the attention mechanism (Bahdanau et al., 2014). It enables the model to find parts of a source sentence that are relevant to predicting a target word (pay attention), without the need to form these parts as a hard segment explicitly. Decoding sentences with the attention-based model resulted in a useful by-product – soft alignments between tokens of source and target sentences. These can be used for many purposes, such as replacing unknown words with back-off translations from a dictionary (Jean et al., 2015) and visualizing the soft alignments (Rikters et al., 2017).

In this paper, we propose using the attention alignments as an indicator of the translation output quality and the confidence of the decoder. We define metrics of confidence that detect and penalize under-translation and over-translation (Tu et al., 2016) as well as input and output tokens with no clear alignment, assuming that all these cases most likely mean that the quality of the translation output is bad.

We apply these attention-based metrics to two use-cases: scoring translations of an NMT system and filtering out the seemingly unsuccessful ones, and comparing translations from two different NMT systems, in order to select the best one.

The structure of this paper is as follows: Section 2 summarizes related work in back-translating with NMT, machine translation combination approaches and confidence estimation. Section 3 introduces the problem of faulty attention distributions and a way to quantify it as a confidence score. Sections 4 and 5 outline the two use-cases for this score – translation filtering and hybrid selections. Finally, we conclude in Section 6 and mention directions for future work in Section 7.

2 Related Work

Back-translation of Monolingual Data

One of the first uses of back-translation of monolingual data as an additional source of training data was reported by (Sennrich et al., 2016a) in their submission for the WMT16 news translation shared task. They translated target-language monolingual corpora into the source language of the respective language pair, and then used the resulting synthetic parallel corpus as additional training data. They performed experiments in ranges from 2 million to 10 million back-translated sentences and reported an increase of 2.2 - 7.7 BLEU (Papineni et al., 2002) for translating between English and Czech, German, Romanian and Russian. The authors also experimented with different amounts of back-translated data and found that adding more data gradually improves performance.

In a later paper Sennrich et al. (2016b) explored other methods of using monolingual data. They experimented with adding an enormous amount of monolingual sentences as targets without any sources to the parallel corpus and compared that to performing back-translation on a part of the monolingual data. While both methods outperform using just parallel data, the back-translated synthetic parallel corpus is a much more powerful addition than the mono data alone.

Pinnis et al. (2017) experimented with using large and even larger amounts of back-translated data and came to a conclusion that any amount is an improvement, but using double the amount gives lower results, while still better than not using any at all. These results hint that it may be possible to get even better results when using only the part of the data selected with some criterion. One of the aims of our work is to provide one such criterion.

Machine Translation System Combination

Zhou et al. (2017) used attention to combine outputs from NMT and SMT systems. The authors first trained intermediate NMT, SMT and hierarchical SMT systems with one-half of the training data. Afterwards, they used each system to translate the target side of the other half of the training data. Finally, the three translated parts as source sentence variants along side the clean target sentence were used for training the combination neural network. This approach gave the network more choices of where to pay attention and which parts should be ignored in the training process. They perform experiments on Chinese→English and report BLEU score improvement by 5.3 points over the best single system and 3.4 points over traditional MT combination methods.

Peter et al. (2016) perform MT system combination in a more traditional manner - using confusion networks. They use 12 different SMT and NMT systems to generate hypothesis translations, align and reorder each hypothesis to match one skeleton hypothesis, creating a confusion network. For the final output is generated by finding the best path in the network. The authors report an improvement of 1.0 BLEU compared to the best single system, translating from English into Romanian.

Translation Confidence Metrics

Lately the idea of modeling coverage in NMT was introduced, for example, Tu et al. (2016) integrate it directly into the attention mechanism and report improved translation quality as a result. On the simpler side of things, Wu et al. (2016b) perform tests with a baseline attention that uses an additional coverage penalty at decoding time; they report no improvement compared to the common length normalization. Our metrics are partially motivated by the coverage penalty, though we apply them at the post-translation stage to determine the confidence of the decoder and the quality of the already made translation, which makes it applicable regardless of which software or approach were used.

Another closely related task is quality estimation. The dominating approach there is collecting post-edits and training a machine learning model to predict the quality score or classify translations into usable/not, near-perfect/not, etc (Bach et al., 2011; Felice and Specia, 2012). The main similarity between our work and quality estimation is their usage of glass-box features (i.e. information about the MT system or the decoder's internal parameters). While our approach does not cover all aspects of quality estimation, it requires no data or training and can be applied to any language and neural machine translation system.

3 Penalizing Attention Disorders

Before describing the confidence metrics based on attention weights, here is a brief overview of the NMT architecture where the attention weights come from.

3.1 Source of Attention

Our work is built around the encoder-decoder machine translation approach (Sutskever et al., 2014; Cho et al., 2014) with an attention mechanism (Bahdanau et al., 2014). In this approach the source tokens are learned to be represented by an encoder, which consists of an embedding layer and a bi-directional LSTM or GRU layer (or 8, Wu et al., 2016b), the outputs of which serve as the learned representation.

There is also a decoder that consists of another layer (or 8, ibid.) of LSTM/GRU cells, with an output layer for predicting the softmax-encoded raw probability distribution of each output word, one at a time. The state of the decoder layer(s) and thus the output distribution depends on the previous recurrent states, the previously produced output word and a weighted sum of the representations of the source sentence tokens. The weights in this sum are generated for every output word by the attention mechanism, which is a feed-forward neural network with the previous state of the decoder and each input word representation as input and the raw weight of that word for the next state as output. Finally, the attention weights are normalized as follows:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$

where e_{ij} is the raw predicted weight and α_{ij} – the final attention weight between the input token j and output token i.

Once the encoder-decoder network has been trained, it can be used to produce translations by predicting the probability for each next word, which can serve as the basis for sampling, greedy search or beam search (Sennrich et al., 2017). We refer the reader for a complete description to (Bahdanau et al., 2014) and ourselves turn on to the main topic of the paper that uses the weights α_{ij} to estimate the confidence of the translations.

Together with the translation, it is also possible to save the attention values between the input tokens and each produced output token. These values can be interpreted as the influence of the input token on the output token, or the strength of the connection between them. Thus, weak or dispersed connections should intuitively indicate a translation with low confidence, while high values and strong connections between one or two tokens on both sides should indicate higher confidence. Next, we present our take at formalizing this intuition.

3.2 Measuring Attention

Figure 1 shows an example of a translation that has little or nothing to do with the input, a frequent occurrence in NMT. Besides the text of the translation, it is clear already by looking at the attention weights of this pair that the translation is weak:

- some input tokens (like the sentence-final full-stop) are most strongly connected to several unrelated output tokens, in other words, their coverage is too high,

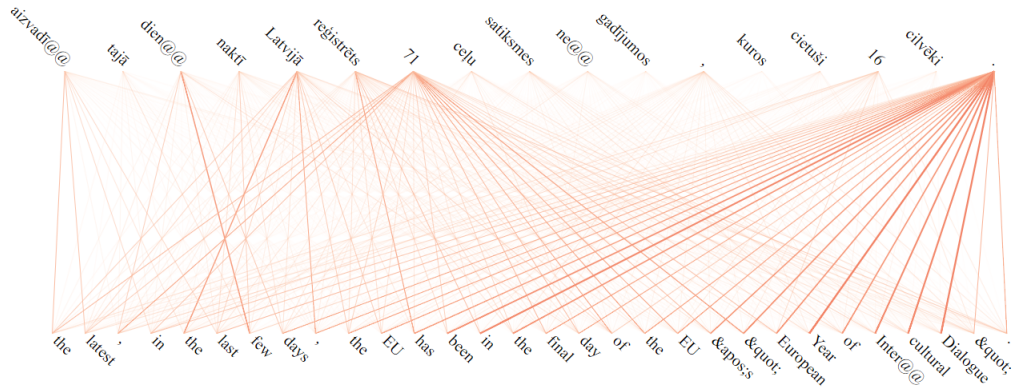


Figure 1: Attention alignment visualization of a bad translation. Reference translation: *71 traffic accidents in which 16 persons were injured have happened in Latvia during the last 24 hours.*, hypothesis translation: *the latest , in the last few days , the EU has been in the final day of the EU 's " European Year of Intercultural Dialogue "*. $CDP = -0.900$, $AP_{out} = -2.809$, $AP_{in} = -2.137$, $Total = -5.846$.

- most of the input token attentions, as well as some output token attentions, are highly dispersed, without one or two clear associations on the counterpart.

On the other hand, a picture like Figure 2 intuitively corresponds to a good translation, with strongly focused alignments. It is this intuition that our metrics formalize: penalizing translations with tokens with a total coverage of not just below but much higher than 1.0, as well as tokens with a dispersed attention distribution.

Coverage Deviation Penalty

Previous work (Wu et al., 2016b) defines a coverage penalty, which is meant to punish translations for not paying enough attention to input tokens:

$$CP = \beta \sum_j \log(\min(\sum_i \alpha_{ji}, 1.0)),$$

where i is the output token index, j – the input token index, β is used to control the influence of the metric and CP – the coverage penalty.

The first part of our metric draws inspiration from the coverage penalty; however, it penalizes not just lacking attention but also too much attention per input token. The aim is to penalize the sum of attentions per input token for going too far from 1.0[1], so tokens with total attention of 1.0 should get a score of 0.0 on the logarithmic scale, while tokens with less attention (like 0.2) or more attention (like 2.5) should get lower values. We thus define the coverage deviation penalty:

$$CDP = -\frac{1}{J} \sum_j \log \left(1 + (1 - \sum_i \alpha_{ji})^2 \right),$$

where J is the length of the input sentence. The metric is on a logarithmic scale, and it is normalized by the length of the input sentence in order to avoid assigning higher scores to shorter sentences[2]. See examples of the CDP metric's values on Figures 1 and 2.

[1]This could be replaced with the token's expected fertility, which we leave for future work

[2]This is not required for choosing translations of the same sentence by the same system, but is required in our

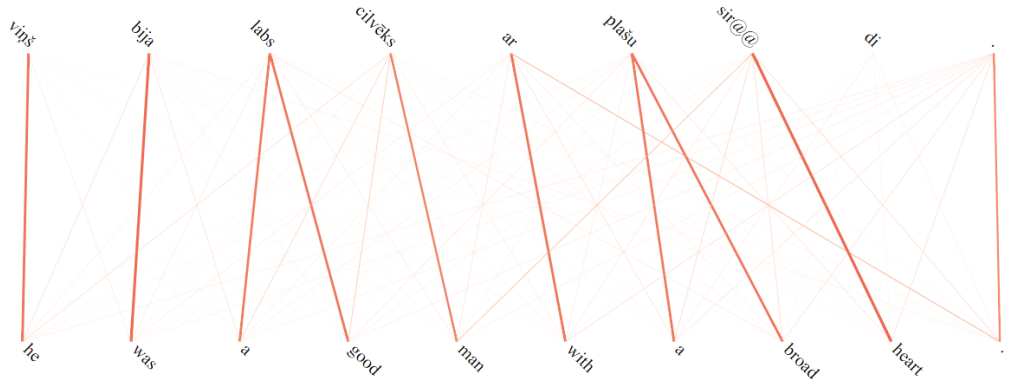


Figure 2: Attention alignment visualization of a good translation. Reference translation: *He was a kind spirit with a big heart.*, hypothesis translation: *he was a good man with a broad heart*. $CDP = -0.099$, $AP_{out} = -1.077$, $AP_{in} = -0.847$, $Total = -2.024$.

Absentmindedness Penalty

However, it is not enough to simply cover the input, we conjecture that more confident output tokens will allocate most of their attention probability mass to one or a small number of input tokens. Thus the second part of our metric is called the absentmindedness penalty and targets scattered attention per output token, where the dispersion is evaluated via the entropy of the predicted attention distribution. Again, we want the penalty value to be 1.0 for the lowest entropy and head towards 0.0 for higher entropies.

$$AP_{out} = -\frac{1}{I}\sum_{i}\sum_{j}\alpha_{ji} \cdot \log \alpha_{ji}$$

The values are again on the log-scale and normalized by the source sentence length I.

The absentmindedness penalty can also be applied to the input tokens after normalizing the distribution of attention per input token, resulting in the counter-part metric AP_{in}. This is based on the assumption that it is not enough to cover the input token, but rather the input token should be used to produce a small number of outputs. See examples of both metric's values on Figures 1 and 2.

Finally, we combine the coverage deviation penalty with both the input and output absentmindedness penalties into a joint metric via summation:

$$confidence = CDP + AP_{out} + AP_{in}$$

Next, we evaluate the metrics directly against human judgments and indirectly by applying them to filtering translations and plugging them into a sentence-level hybrid translation scheme.

3.3 Human Evaluation

It is clear that the defined metrics only paint a partial picture, since they rely on the attention weights only. For instance, they do not evaluate the lexical correspondence between the source and hypothesis, and more generally, being confident does not mean being right. We wanted to find out how much confidence in our case correlates with translation quality.

experiments described in the next sections.

To do so we asked human volunteers to perform pairwise ranking of translations from two baseline NMT systems: one done with Nematus (Sennrich et al., 2017) and the other – with Neural Monkey (Helcl and Libovickỳ, 2017). The translations and measurements were done for English-Latvian and Latvian-English, using corpora from the news translation shared task of WMT'2017; further details can be found in Section 4. We selected 200 random sentences for both translation directions and these were given to native Latvian speakers for evaluation. The MT-EQuAl (Girardi et al., 2014) tool was used for the evaluation task. The evaluators were shown one source sentence at a time along with the two different translations. They were instructed to assign one of five categories for each translation: "worst", "bad", "ok", "good" or "best", noting that both may be categorized as equally "good" or "bad", etc. Differing judgments for the same sentence were averaged. All 200 sentences were annotated by at least one human annotator.

It makes more sense to treat the results as relative comparisons, not absolute scores, as the annotators only see two translations at a time. We use these comparisons to compute the Kendall rank correlation coefficient (Kendall, 1938) by only looking at the pairs where human scores differ. Since we only have comparisons for each pair and not between different sentences, the coefficient is computed as

$$\tau = \frac{pos - neg}{pos + neg},$$

where pos is the number of pairs where the metric agrees with the human judgment and neg is the number of pairs where they disagree.

The results are presented in Table 1, and as we can see they indicate weak correlation, with the absolute values of τ between 0.012 and 0.200.

Language pair	CDP	AP_{in}	AP_{out}	Overall
En→Lv	0.099	0.074	0.123	0.086
Lv→En	-0.012	-0.153	-0.200	-0.153

Table 1: The Kendall's Tau correlation between human judgments and the confidence scores.

Let us look closer at where the metrics disagree with human judgments. Figure 3 shows an example of a translation which was rated highly by human annotators but poorly with our metrics. While the sentence is a good translation, it does not follow the source word-by-word. Some subword units and functional words do not have a clear alignment, even though they are understood/generated correctly. This means that one problem with our metrics is that they might be over-penalizing translations that deviate from a direct literal translation.

Next, we continue with the experiments of using our metrics to filter synthetic data and to select translations in a hybrid MT scenario.

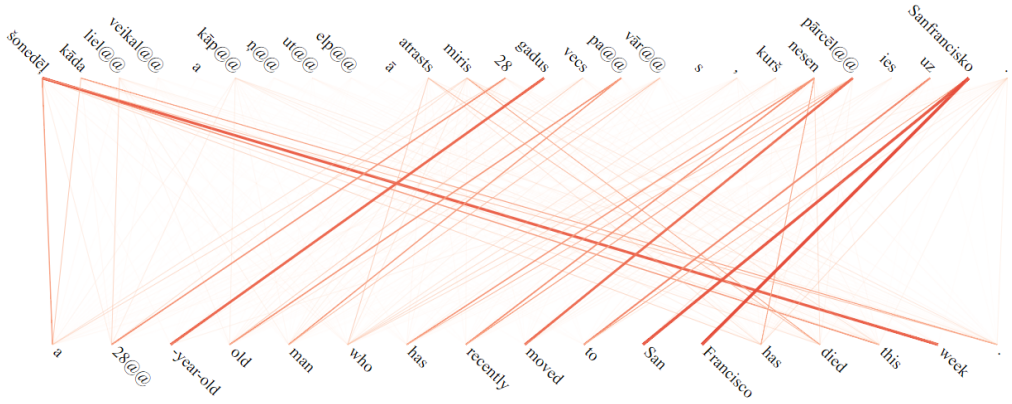


Figure 3: Attention alignment visualization of a bad translation. Reference translation: *a 28-year-old chef who had recently moved to San Francisco was found dead in the stairwell of a local mall this week .*, hypothesis translation: *a 28-year-old old man who has recently moved to San Francisco has died this week .*, $CDP = -0.250$, $AP_{out} = -1.740$, $AP_{in} = -1.46$, $Total = -3.45$.

4 Filtering Back-translated Data

4.1 Baseline Systems and Data

Our baseline systems were trained with two NMT frameworks - Nematus (NT) (Sennrich et al., 2017) and Neural Monkey (NM) (Helcl and Libovickỳ, 2017). For all NMT models we used a shared subword unit vocabulary (Sennrich et al., 2016c) of 35000 tokens, clip the gradient norm to 1.0 (Pascanu et al., 2013), dropout of 0.2, trained the models with Adadelta (Zeiler, 2012) and performed early stopping after 7 days of training. For models with each NMT framework we used the default settings as mentioned in the frameworks documentation:

- For NT models we used a maximum sentence length of 50, word embeddings of size 512, and hidden layers of size 1000. For decoding with NT we used beam search with a beam size of 12.
- For NM models we used a maximum sentence length of 70, word embeddings and hidden layers of size 600. For decoding with NM a greedy decoder was used.

Training, development and test data for all systems in both language pairs and translation directions was used from the WMT17 news translation task [3]. For the baseline systems, we used all available parallel data, which is 5.8 million sentences for En↔De and 4.5 million sentences for En↔Lv.

4.2 Back-translating and Filtering

We used our baseline En→Lv and Lv→En NM and NT systems to translate all available Latvian monolingual news domain data - 6.3 million sentences in total from *News Crawl: articles from 2014, 2015, 2016*, and the first 6 million sentences from the English *News Crawl 2016*. Much more monolingual data was available from other domains aside from news. Since the development and test data was of the news domain, we only used that, considering it as in-domain data for our systems.

[3] EMNLP 2017 Second Conference on Machine Translation - http://www.statmt.org/wmt17/

For each translation, we used the attention provided from the NMT system to calculate our confidence score, sorted all translations according to the score and selected the top half of the translations along with the corresponding source sentences as the synthetic parallel corpus. We used only the full confidence score (combination of CDP, AP_{out} and AP_{in}) for filtering instead of each individual score due to its smoother overall correlation with human judgments. In between, we also removed any translation that contained any <*unk*> tokens.

To compare attention-based filtering with a different method, we trained a CharRNN[4] language model (LM) with 4 million news sentences from each of the target languages. We used these LMs to get perplexity scores for all translations, order them and get the *better half*. Table 2 summarizes how much human evaluation overlaps with each of the filtering methods. The final row indicates how much both filtering methods overlap with each other. While results from either approach don't look overly convincing, the LM-based approach has been proven to correlate with human judgments close to the BLEU score and is a good evaluation method for MT without reference translations (Gamon et al., 2005). Therefore the attention-based approach that does not require training of an additional model and overlaps with human judgments to approximately the same level should be more desirable.

Filtering Method	En→Lv	Lv→En
LM-based overlap with human	58%	56%
Attention-based overlap with human	52%	60%
LM-based overlap with Attention-based	34%	22%

Table 2: Human judgment overlap results on 200 random sentences from the *newsdev2017* dataset compared to filtering methods.

4.3 NMT with Filtered Synthetic Data

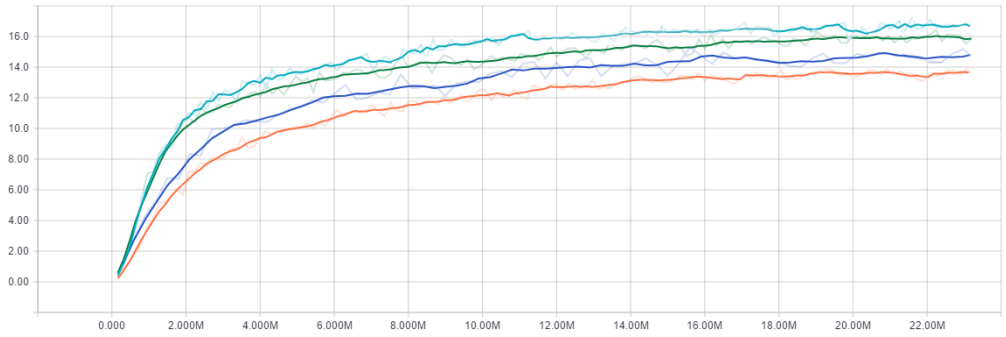


Figure 4: Automatic evaluation progression of Lv→En experiments on validation data. Orange – baseline; dark blue — with full back-translated data; green – with LM-filtered back-translated data; light blue – with attention-filtered back-translated data.

We shuffled each synthetic parallel corpus with the baseline parallel corpora and used them to train NMT systems. In addition to the baseline and two types of filtered BT synthetic data, we also trained a system with the full BT data for each translation direction. Figure 4

[4]Multi-layer Recurrent Neural Networks (LSTM, GRU, RNN) for character - level language models in Torch https://github.com/karpathy/char-rnn

shows a combined training progress chart for Lv→En on the full *newsdev2017* dataset that was used as the development set for training. Here the differences between all four approaches are clearly visible. Further results on a subset of *newsdev2017* and the full *newstest2017* dataset are summarized in Table 3. While for Lv→En and En↔De the attention-based approach is the clear leader, for En→Lv it falls behind the LM filtered version. We were not able to identify a clear reason for this and leave it for the future work. As expected, adding BT synthetic training data allows to get higher BLEU scores in all cases. It can be observed that filtering out half of the badly translated data and keeping only the best translations either does not decrease the final output quality in some cases or even further increase the quality in others, when using the LM. With filtering by attention, the results are more inconsistent - even higher in one direction while deterioration in the other. A reason for this could be that for Lv→En attention-based filtering the similarity with human judgments was higher than for En→Lv (Table 2), and it was also more different from the LM-based one. While for the other direction it is the other way around.

	BLEU							
Dataset	Dev	Test	Dev	Test	Dev	Test	Dev	Test
System	En→Lv		Lv→En		En→De		De→En	
Baseline	8.36	11.90	8.64	12.40	25.84	20.11	30.18	26.26
+ Full Synthetic	9.42	13.50	9.01	13.81	28.97	22.68	34.82	29.35
+ LM-Filtered Synthetic	9.75	13.52	9.45	14.30	29.59	23.48	34.47	29.42
+ Attn.-Filtered Synth.	8.99	12.76	**11.23**	**14.83**	**30.19**	23.16	**35.19**	**29.47**

Table 3: Experiment results in BLEU for translating between English↔Latvian with different types of back-translated data using development (200 random sentences from *newsdev2017*) and test (*newstest2017*) datasets.

5 Attention-based Hybrid Decisions

We translated the development set with both baseline systems for each language pair in each direction. The hybrid selection of the best translation was performed similarly to filtering, where we discarded the worst-scoring half of the translations. In the hybrid selection, we used the same score to compare both translations of a source sentence and choose the better one. Results of the hybrid selection experiments are summarized in Table 4. For translating between En↔Lv, where the difference between the baseline systems is not that high (0.06 and 1.55 BLEU), the hybrid method achieves some meaningful improvements. However, for En↔De, where differences between the baseline systems are bigger (3.46 and 4.46 BLEU), the hybrid drags both scores down.

	BLEU			
System	En→De	De→En	En→Lv	Lv→En
Neural Monkey	18.89	26.07	13.74	11.09
Nematus	22.35	30.53	13.80	12.64
Hybrid	20.19	27.06	**14.79**	**12.65**
Human	23.86	34.26	15.12	13.24

Table 4: Hybrid selection experiment results in BLEU on the development dataset (200 random sentences from *newsdev2017*).

The last row of the results Table 4 shows BLEU scores for the scenario when human an-

notator preferences were used to select each output sentence. An overview of human evaluator preferred translation selections is visible in Table 5. The results show that out of all translations the human evaluators deliberately prefer one or the other system. Aside from En→Lv, where a slight tendency towards Neural Monkey translations can be observed, all others look more or less equal. This highly contrasts with the BLEU scores from Table 4, where in both translation directions from English human evaluators prefer the lower-scoring system more often than the higher-scoring one. The final row of Table 5 shows how much our attention-based score matches the human judgments in selecting the best translation.

System	En→De	De→En	En→Lv	Lv→En
Neural Monkey	54%	42%	61.5%	47%
Nematus	46%	58%	38.5%	53%
Overlaps with hybrid selection	57%	47%	62.5%	51%

Table 5: Human evaluation results on 200 random sentences from the *newsdev2017* dataset compared to attention-hybrid selection.

6 Conclusions

In this paper, we described how attentional data from neural machine translation systems can be useful for more than just visualizations or replacing specific tokens in the output. We introduced an attention-based confidence score that can be used for judging NMT output. Two applications of using attentional data were investigated and compared to similar approaches. We used a smaller dataset to perform manual evaluation and compared that to all automatically obtained results. Our experiments showed interesting results and some increases in automated evaluation, as well as a good correlation with human judgments.

In addition to the methods described in this paper, we release open-source scripts[5] for (1) scoring, ordering and filtering NMT translations, (2) performing hybrid selections between two different NMT outputs of the same source, and (3) software for inspecting attention alignments that the NMT systems produce in the translation process (used for Figures 1 and 2). We also provide all development subsets that we used for manual evaluation with anonymized human annotations.

7 Future Work

This paper introduced the first steps in using NMT attention for less obvious intentions. It seemed that the attention score can complement the LM perplexity score in distinguishing good from bad translations. An idea for future experiments could be combining these scores to achieve a higher correlation with human judgments.

Additional improvements can be made to the hybrid decisions as well. Since the score represents the systems *confidence*, a badly trained NMT system can be more confident about a bad translation than a good system about a decent translation. While a hybrid combination of two similar quality NMT systems did put the attention score to good use, in the case with different quality systems the confidence of the weaker one was a pitfall. This indicates that the confidence score could be used in ensemble with a quality estimation score or used as a feature in training an MT quality estimation system.

For filtering synthetic back-translated data we dropped the worst-scoring 50% of the data, but this threshold may not be optimal for all scenarios. Several paths worth more exploration

[5]Confidence Through Attention - https://github.com/M4t1ss/ConfidenceThroughAttention

include exploring the effects of different static thresholds (e.g. 30% or 70%) or clustering the data by confidence score and dropping the lowest-scoring one or two clusters. Another path worth exploring for filtering would be to see how filtering by each individual score (CDP, AP_{in}, AP_{out}) compares to filtering by confidence.

In the near future, we also plan to supplement an attention inspection tool so that it displays confidence metrics and additional visualizations based on these scores.

References

Bach, N., Huang, F., and Al-Onaizan, Y. (2011). Goodness: A method for measuring machine translation confidence. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 211–219, Portland, Oregon, USA. Association for Computational Linguistics.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Felice, M. and Specia, L. (2012). Linguistic features for quality estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 96–103, Montréal, Canada.

Gamon, M., Aue, A., and Smets, M. (2005). Sentence-level mt evaluation without reference translations: Beyond language modeling. In *Proceedings of EAMT*, pages 103–111.

Girardi, C., Bentivogli, L., Farajian, M. A., and Federico, M. (2014). MT-EQuAl: a Toolkit for Human Assessment of Machine Translation Output. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 120–123.

Helcl, J. and Libovickỳ, J. (2017). Neural monkey: An open-source tool for sequence learning. *The Prague Bulletin of Mathematical Linguistics*, 107(1):5–17.

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2015). On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China. Association for Computational Linguistics.

Kendall, M. (1938). A new measure of rank correlation. *Biometrika*, 30(1-2):81–89.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318.

Peter, J.-T., Alkhouli, T., Ney, H., Huck, M., Braune, F., Fraser, A., Tamchyna, A., Bojar, O., Haddow, B., Sennrich, R., Blain, F., Specia, L., Niehues, J., Waibel, A., Allauzen, A., Aufrant, L., Burlot, F., Knyazeva, E., Lavergne, T., Yvon, F., Frank, S., Daiber, J., and Pinnis, M. (2016). The QT21/HimL Combined Machine Translation System. *Proceedings of the First Conference on Machine Translation (WMT 2016), Volume 2: Shared Task Papers*, 2:344—-355.

Pinnis, M., Krislauks, R., Deksne, D., and Miks, T. (2017). Neural machine translation for morphologically rich languages with improved sub-word units and synthetic data. In *International Conference on Text, Speech, and Dialogue*, pages 20–27. Springer.

Rikters, M., Fishel, M., and Bojar, O. (2017). Visualizing neural machine translation attention and confidence. *The Prague Bulletin of Mathematical Linguistics*, 109(3):in print.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli Barone, A. V., Mokry, J., and Nadejde, M. (2017). Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016c). Neural Machine Translation of Rare Words with Subword Units. In *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, Montreal, Canada.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016). Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016a). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016b). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhou, L., Hu, W., Zhang, J., and Zong, C. (2017). Neural System Combination for Machine Translation.

Disentangling ASR and MT Errors in Speech Translation

Ngoc-Tien Le ngoc-tien.le@univ-grenoble-alpes.fr
Benjamin Lecouteux benjamin.lecouteux@univ-grenoble-alpes.fr
Laurent Besacier laurent.besacier@univ-grenoble-alpes.fr
University Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France

Abstract

The main aim of this paper is to investigate automatic quality assessment for spoken language translation (SLT). More precisely, we investigate SLT errors that can be due to transcription (ASR) or to translation (MT) modules. This paper investigates automatic detection of SLT errors using a single classifier based on joint ASR and MT features. We evaluate both 2-class ($good$/bad) and 3-class ($good$/bad_{ASR}/bad_{MT}) labeling tasks. The 3-class problem necessitates to disentangle ASR and MT errors in the speech translation output and we propose two label extraction methods for this non trivial step. This enables - as a by-product - qualitative analysis on the SLT errors and their origin (are they due to transcription or to translation step?) on our large in-house corpus for French-to-English speech translation.

Index Terms: Spoken Language Translation, Automatic Speech Recognition, Confidence Estimation, Quality Estimation, ASR and MT errors detection.

1 Introduction

This paper addresses a relatively new quality assessment task: error detection in spoken language translation (SLT) using both automatic speech recognition (ASR) features and machine translation (MT) features. To our knowledge, the first attempts to design error detection for speech translation, using both ASR and MT features, are our own work (Besacier et al., 2014, 2015) which is further extended in this paper submission.

Contributions (1) This paper extends previous work (Besacier et al., 2014, 2015) in 2-class ($good$/bad) error detection in SLT using a single classifier based on joint ASR and MT features (2) in order to disentangle ASR and MT errors in SLT, we extend error detection to a 3-class problem ($good$/bad_{ASR}/bad_{MT}) where we try to find the source of the SLT errors (3) two methods are compared for setting such 3-class labels on our corpus and a first attempt to automatically detect errors and their origin in a SLT output is presented at the end of this paper.

Outline The outline of this paper goes simply as follows: Section 2 formalizes error detection in SLT and presents our experimental setup. Section 3 proposes two methods to disentangle ASR and MT errors in SLT output and presents statistics on a large French-English corpus. Section 4 presents our 2-class and 3-class error detection results while section 5 concludes this work and gives some perspectives.

2 Automatic Error Detection in Speech Translation

2.1 Formalization

A quality estimation (or error detection) component in speech translation solves the equation:

$$\hat{q} = \underset{q}{\operatorname{argmax}}\{p_{SLT}(q|x_f, f, \hat{e})\} \quad (1)$$

where x_f is the given signal in the source language; $\hat{e}$ [1] $= (e_1, e_2, ..., e_N)$ is the most probable target language sequence from the spoken language translation (SLT) process; $f = (f_1, f_2, ..., f_M)$ is the transcription of x_f; $q = (q_1, q_2, ..., q_N)$ is a sequence of error labels on the target language and $q_i \in \{good, bad\}$ [2]. This is a sequence labeling task that can be solved with several machine learning techniques such as Conditional Random Fields (CRF) (Lafferty et al., 2001). However, for that, we need a large amount of training data for which a quadruplet (x_f, f, e, q) is available.

As it is much easier to obtain data containing either the triplet (x_f, f, q) (ASR output + manual references and error labels inferred from WER) or the triplet (f, e, q) (MT output + manual post-editions and error labels inferred using tools such as TERp-A (Snover et al., 2008)) we can also recast error detection with the following equation:

$$\hat{q} = \underset{q}{\operatorname{argmax}}\{p_{ASR}(q|x_f, f)^{\alpha} * p_{MT}(q|e, f)^{1-\alpha}\} \quad (2)$$

where α is a weight giving more or less importance to error detector on transcription compared to error detector on translation.

2.2 Dataset, ASR and MT Modules

2.2.1 Dataset

In this paper, we use our in-house corpus made available on a *github* repository [3] for reproductibility. The *dev* set and *tst* set of this corpus were recorded by french native speakers. Each sentence was uttered by 3 speakers, leading to 2643 and 4050 speech recordings for *dev* set and *tst* set, respectively. For each speech utterance, a quintuplet containing: ASR output (f_{hyp}), verbatim transcript (f_{ref}), text translation output ($e_{hyp_{mt}}$), speech translation output ($e_{hyp_{slt}}$) and post-edition of translation (e_{ref}) is available. The total length of the union of *dev* and *tst* is 16h52 (42 speakers - 5h51 for *dev* and 11h01 for *tst*).

2.2.2 ASR Systems

To obtain the speech transcripts (f_{hyp}), we built a French ASR system based on KALDI toolkit (Povey et al., 2011). Acoustic models are trained using several corpora (ESTER, REPERE, ETAPE and BREF120) representing more than 600 hours of french transcribed speech. We use two 3-gram language models trained on French ESTER corpus (Galliano et al., 2006) as well as on French Gigaword (vocabulary size are respectively 62k and 95k). ASR systems LM weight parameters are tuned through WER on *dev* corpus. *Table 1* presents the performances obtained by both ASR systems.

2.2.3 SMT System

We used *moses* phrase-based translation toolkit (Koehn et al., 2007) to translate French ASR into English (e_{hyp}). This medium-size system was trained using a subset of data provi-

1. written simply e for convenience in any other equations
2. at this point q_i takes two values (G/B) but will evolve to 3 labels later on in section 3
3. `https://github.com/besacier/WCE-SLT-LIG/`

ded for IWSLT 2012 evaluation (Federico et al., 2012): Europarl, Ted and News-Commentary corpora. The total amount is about 60M words. We used an adapted target language model trained on specific data (News Crawled corpora) similar to our evaluation corpus (see (Potet et al., 2010)).

2.3 Obtaining Error Labels for SLT

After building an ASR system, we have a new element of our desired quintuplet: the ASR output f_{hyp}. It is the noisy version of our already available verbatim transcripts called f_{ref}. This ASR output (f_{hyp}) is then translated by the SMT system (Potet et al., 2010) already mentioned in subsection 2.2.3. This new output translation is called $e_{hyp_{slt}}$ and it is a degraded version of $e_{hyp_{mt}}$ (translation of f_{ref}). To infer the quality (G, B) labels of our speech translation output $e_{hyp_{slt}}$, we use TERp-A toolkit (Snover et al., 2008) between $e_{hyp_{slt}}$ and e_{ref} (more details can be found in our former paper (Besacier et al., 2015)). Table 1 summarizes baseline ASR, MT and SLT performances obtained on our corpora, as well as the distribution of $good$ (G) and bad (B) labels inferred for both tasks. Logically, the percentage of (B) labels increases from MT to SLT task in the same conditions and it decreases when ASR system improves.

Task	***ASR (WER)***		***MT (BLEU)***		***% G (good))***		***% B (bad)***	
	dev set	*tst* set	*dev* set	*tst* set	*dev* set	*tst* set	*dev* set	*tst* set
MT			49.13%	57.87%	76.93%	81.58%	23.07%	18.42%
SLT (ASR1)	21.86%	17.37%	26.73%	36.21%	62.03%	70.59%	37.97%	29.41%
SLT (ASR2)	16.90%	12.50%	28.89%	38.97%	63.87%	72.61%	36.13%	27.39%

Table 1. ASR, MT and SLT performances on our *dev* set and *tst* set.

3 Disentangling ASR and MT Errors

In previous section, we only extract $good/bad$ labels from the SLT output while it might be interesting to move from a 2-class problem to a 3-class problem in order to label our SLT hypotheses with one of the 3 following labels: $good$ (G), *asr-error* (*B_ASR*) and *mt-error* (*B_MT*). Before training automatic systems for error detection, we need to set such 3-class labels on our dev and $test$ corpora. For that, we propose, in the next sub-sections, two slightly different methods to extract them. The first one is based on word alignments between SLT and MT and the second one is based on a simpler SLT-MT error subtraction.

3.1 Method 1 - Word Alignments between MT and SLT

In machine translation, fertility of a source word designs to how many output words it translates. If we transpose this definition to our disentangling problem, then *fertility of an MT error* designs how many erroneous words - in the SLT output - it is aligned to. From this simple definition, we derive our first way (*Method 1*) to generate 3-class annotations.

Let $\hat{e}_{slt} = (e_1, e_2, \ldots, e_n)$: the set of SLT hypotheses ($e_{hyp_{slt}}$); e_{k_j} denotes the j^{th} word in the sentence e_k, where $1 \leq k \leq n$

Let $\hat{e}_{mt} = (e'_1, e'_2, \ldots, e'_n)$: the set of MT hypotheses ($e_{hyp_{mt}}$); e'_{k_i} denotes the i^{th} word in the sentence e'_k, where $1 \leq k \leq n$

Let $L = (l_1, l_2, \ldots, l_n)$: the set of the word alignments from sentences in $e_{hyp_{slt}}$ to related sentences in $e_{hyp_{mt}}$, where l_k contains the word alignments from sentence e_k to relevant sentence e'_k, $1 \leq k \leq n$; (e_{k_j}, e'_{k_i}) = ***True***, if there is one word alignment between e_{k_j} and e'_{k_i}; (e_{k_j}, e'_{k_i}) = ***False***, otherwise.

Our algorithm for *Method 1* is defined as *Algorithm 1*. This method relies on word alignments and uses MT labels. We also propose a simpler method in the next section.

Algorithm 1 *Method 1* - Using word alignments between MT and SLT

$list\_labels\_result \leftarrow empty\_list$
for each sentence $e_k \in \hat{e}_{slt}$ **do**
  $list\_labels\_sent \leftarrow empty\_list$
  for $j \leftarrow 1$ to $NumberOfWords(e_k)$ **do**
    if $label(e_{k_j}) =$ 'G' **then**
      add 'G' to $list\_labels\_sent$
    else if Existed Word Alignment (e_{k_j}, e'_{k_i}) and $label(e'_{k_i})$='B' **then**
      add 'B_MT' to $list\_labels\_sent$
    else
      add 'B_ASR' to $list\_labels\_sent$
    end if
  end for
  add $list\_labels\_sent$ to $list\_labels\_result$
end for

3.2 Method 2 - Subtraction between SLT and MT Errors

Our second way to extract 3-class labels (*Method 2*) focuses on the differences between SLT hypothesis ($e_{hyp_{slt}}$) and MT hypothesis ($e_{hyp_{mt}}$). We call it *subtraction between SLT and MT errors* because we simply consider that errors present in SLT and not present in MT are due to ASR. This method has a main difference with the previous one: it does not rely on the extracted labels for MT.

Our intuition is that the number of *mt-error*s estimated will be slightly lower than for *Method 1* since we first estimate the number of *asr-error*s and the rest is considered - by default - as *mt-error*s.

With the same notations of *Method 1*, but highlighting that $L = (l_1, l_2, \ldots, l_n)$ is the set of alignments through edit distance between $e_{hyp_{slt}}$ and $e_{hyp_{mt}}$, where l_{k_i} corresponds to "Insertion", "Substitution", "Deletion" or "Exact". Our algorithm for *Method 2* is defined as follows.

3.3 Example with 3-label Setting

Table 2 gives the edit distance between a SLT and MT hypothesis while table 3 shows how *Method 1* and *Method 2* set 3-class labels to the SLT hypothesis. One transcript (f_{hyp}) has 1 error. This drives 3 B labels on SLT output ($e_{hyp_{slt}}$), while $e_{hyp_{mt}}$ has only 2 B labels. As can be seen in the cases of *Method 1* and *Method 2*, we respectively have (1 B_ASR, 2 B_MT) and (2 B_ASR, 1 B_MT).

$e_{hyp_{slt}}$	surgeons	in	los	angeles	it	is	said
$e_{hyp_{mt}}$	surgeons	in	los	angeles	**	have	said
edit op.	Exact	Exact	Exact	Exact	Insertion	Substitution	Exact

Table 2. Example of edit distance between SLT and MT.

Algorithm 2 *Method 2* - Subtraction between SLT and MT errors

$list\_labels\_result \leftarrow empty\_list$
for each sentence $e_k \in \hat{e}_{slt}$ **do**
  $list\_labels\_sent \leftarrow empty\_list$
  for $j \leftarrow 1$ to $NumberOfWords(e_k)$ **do**
    if $label(e_{k_j}) =$ 'G' **then**
      add 'G' to $list\_labels\_sent$
    else if $NameOfWordAlignment(l_{k_i})$ is 'Insertion' OR 'Substitution' **then**
      add 'B_ASR' to $list\_labels\_sent$
    else
      add 'B_MT' to $list\_labels\_sent$
    end if
  end for
  add $list\_labels\_sent$ to $list\_labels\_result$
end for

f_{ref}	les chirurgiens	de	los	angeles	ont		dit
f_{hyp}	les chirurgiens	de	los	angeles	on		dit
labels ASR	G G	G	G	G	B		G
$e_{hyp_{mt}}$	surgeons	in	los	angeles		have	said
labels MT	G	B	G	G		B	G
$e_{hyp_{slt}}$	surgeons	in	los	angeles	it	is	said
labels SLT (2-label)	G	B	G	G	B	B	G
labels SLT (*Method 1*)	G	B_MT	G	G	B_ASR	B_MT	G
labels SLT (*Method 2*)	G	B_MT	G	G	B_ASR	B_ASR	G
e_{ref}	the surgeons	of	los	angeles			said

Table 3. Example of quintuplet with 2-label and 3-label.

These differences are due to slightly different algorithms for label extraction. As Table 3 presents, "is" (SLT hypothesis) is aligned to "have" (MT hypothesis) and "have" (MT hypothesis) is labeled by "B". It can therefore be assumed that "is" (SLT hypothesis) should be annotated with word-level labels by B_MT according to *Method 1*. However, using *Method 2*, "is" (SLT hypothesis) could be labeled by B_ASR because the type of word alignment between "is" (SLT hypothesis) and "have" (MT hypothesis) is substitution (S), as shown in Table 2.

3.4 Statistics with 3-label Setting on the Whole Corpus

Table 4 presents the summary statistics for the distribution of *good* (G), *asr-error* (B_ASR) and *mt-error* (B_MT) labels obtained with both label extraction methods. We see that both methods give similar statistics but slightly different rates of B_ASR and B_MT.

As can be seen from Table 4, it is interesting to note that while ASR system improves from *ASR1* to *ASR2*, the rate of B_ASR labels logically decreases by more than 2 points, while the rate of B_MT remains almost stable (less than 1 point difference) which makes sense since the MT system is the same in both *ASR1* and *ASR2*. These statistics show that intersection between both methods is probably a good estimation of disentangled ASR and MT errors in SLT.

Task - *ASR1*	*dev* set			*tst* set		
	%G	*%B_ASR*	*%B_MT*	%G	%B_ASR	%B_MT
label/m1:Method 1	*62.03*	*19.09*	*18.89*	70.59	14.50	14.91
label/m2:Method 2	*62.03*	*22.49*	*15.49*	70.59	16.62	12.79
label/same(m1, m2)	*62.03*	*18.09*	*14.49*	70.59	13.58	11.88
label/diff(m1, m2)	*0*	*1.00*	*4.40*	0	0.92	3.03
Task - *ASR2*	***dev* set**			***tst* set**		
	%G	***%B_ASR***	***%B_MT***	**%G**	**%B_ASR**	**%B_MT**
label/m1:Method 1	*63.87*	*16.89*	*19.23*	72.61	11.92	15.47
label/m2:Method 2	*63.87*	*19.78*	*16.34*	72.61	13.58	13.81
label/same(m1, m2)	*63.87*	*16.05*	*15.50*	72.61	11.12	13.01
label/diff(m1, m2)	*0*	*0.84*	*3.73*	0	0.80	2.46

Table 4. Statistics with 3-label setting for *ASR1* and *ASR2*.

3.5 Qualitative Analysis of SLT Errors

Our new 3-label setting procedure allows us to analyze the behavior of our SLT system.

f_{ref}	peter frey est né le quatre août mille neuf cent cinquante sept à bingen
f_{hyp_1}	**pierre ferait aimé** le quatre août mille neuf cent cinquante sept à **big m**
f_{hyp_2}	**pierre** frey est né le quatre août mille neuf cent cinquante sept à **big m**
$e_{hyp_{mt}}$	peter frey was born on **4** august 1957 **to** bingen .
$e_{hyp_{slt1}}$	**pierre would liked the four** august **thousand nine hundred and fifty seven to big m**
$e_{hyp_{slt2}}$	**pierre** frey is born **the four** august **thousand nine hundred and fifty seven to big m**
e_{ref}	peter frey was born on august 4th 1957 in bingen .

Table 5. Example 1 - SLT hypothesis annotated with two methods - having a few *asr-errors*, a few *mt-errors* and many *slt-errors* such as 5 B_ASR1, 3 B_ASR2, 2 B_MT, 14 B_SLT1, 12 B_SLT2.

We can observe sentences with Table 5 presents, as an example, few ASR and MT errors leading to many SLT errors. Indeed, this is a good way of detecting flaws in the SLT pipeline such as bad post-processing of the SLT output (numerical or text dates, for instance).

As shown in Table 6, on the contrary, there are many ASR errors leading to few SLT errors (ASR errors with few consequences such as morphological substitutions - for instance in French: de/des, déficit/déficits, budgétaire/budgétaires).

Finally, ASR errors as presented in Table 7 have different consequences on SLT quality (on a sample sentence, 2 ASR errors of system 1 and 2 lead to 14 and 9 SLT errors, respectively).

Figure 1 shows how our speech utterances are distributed in the two-dimensional (B_{ASR}, B_{MT}) error space.

4 Automatic Error Detection for SLT

In this paper, we use Conditional Random Fields (Lafferty et al., 2001) (CRFs) as our machine learning method, with WAPITI toolkit (Lavergne et al., 2010), to train our error detector based on MT and ASR engineered features. For ASR, we extract 9 features, which come from the ASR graph, from language model scores and from a morphosyntactic analysis. These detai-

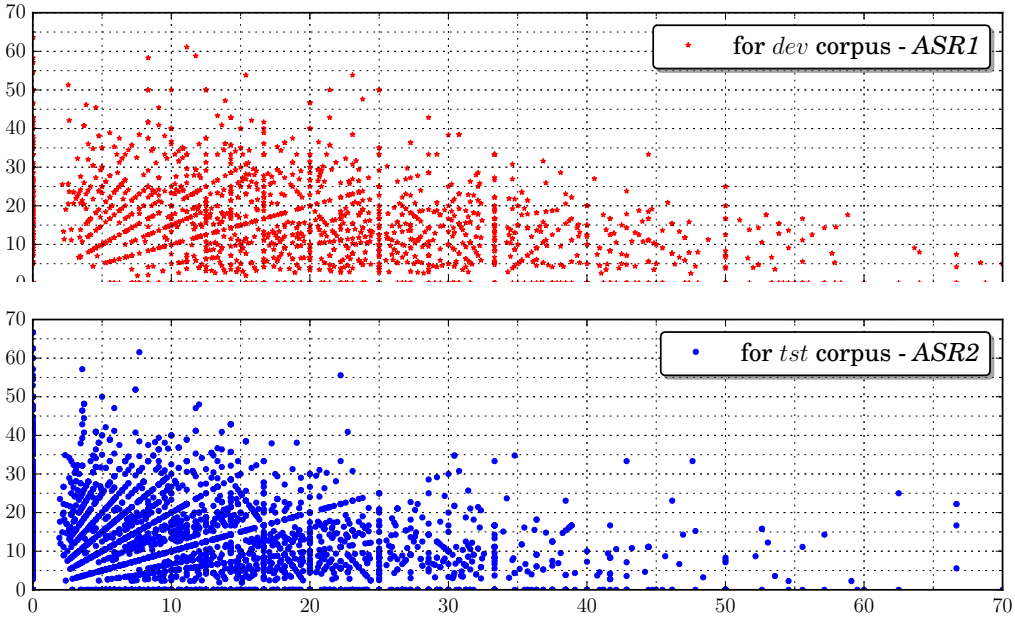


Figure 1. Example of the rate (%) of ASR errors (x-axis) versus (%) MT errors (y-axis) - for *dev/ASR1* and *tst/ASR2*.

f_{ref}	malheureusement le système européen de financement gouvernemental direct est
f_{hyp_1}	malheureusement le système européen financement **gouvernementale directe et**
f_{hyp_2}	malheureusement le système européen de financement gouvernemental direct est
$e_{hyp_{mt}}$	unfortunately , the european system of direct government funding is
$e_{hyp_{slt1}}$	unfortunately the european system direct government funding
$e_{hyp_{slt2}}$	unfortunately the european system of direct government funding is
e_{ref}	unfortunately , the european system of direct government funding is
f_{ref}	victime de la croissance économique européenne lente et des déficits budgétaires
f_{hyp_1}	**victimes** de la croissance économique européenne **venant de déficit budgétaire**
f_{hyp_2}	victime de la croissance économique européenne **venant** des déficits budgétaires
$e_{hyp_{mt}}$	a victim of european economic growth **slow** and budget deficits .
$e_{hyp_{slt1}}$	**and** victims of european economic growth **from** budget deficit
$e_{hyp_{slt2}}$	a victim of european economic growth **from the** budget deficits
e_{ref}	a victim of slow european economic growth and budget deficits .

Table 6. Example 2 - SLT hypothesis annotated with two methods - having many *asr-errors*, a few *mt-errors* and a few *slt-errors* such as 8 B_ASR1, 1 B_ASR2, 1 B_MT, 2 B_SLT1, 2 B_SLT2.

f_{ref}	nous ne comprenons pas ce qui se passe chez les jeunes pour qu' ils trouvent
f_{hyp_1}	nous ne comprenons pas **ceux** qui se passe chez les jeunes pour qu' ils trouvent
f_{hyp_2}	nous ne comprenons pas ce qui se passe chez les jeunes pour qu' **il trouve**
$e_{hyp_{mt}}$	we do not understand what is happening **among** young people for **that**
$e_{hyp_{slt1}}$	we do not understand **those who happens among** young people for **that**
$e_{hyp_{slt2}}$	we do not understand what is happening **among** young people
e_{ref}	we do not understand what is happening in young people 's mind for them
f_{ref}	amusant de maltraiter gratuitement un animal sans défense qui nous donne
f_{hyp_1}	amusant de **maltraité** gratuitement un animal sans défense qui nous
f_{hyp_2}	amusant de maltraiter gratuitement un animal sans défense qui nous donne
$e_{hyp_{mt}}$	**they are fun** to mistreat **free a** defenceless animal
$e_{hyp_{slt1}}$	**they** find **fun free** mistreated **a** defenceless animal
$e_{hyp_{slt2}}$	to find **it** amusing to mistreat **free a** defenceless animal
e_{ref}	to find amusing to mistreat defenceless animals without reason ,
f_{ref}	de l' affection de l' amitié et nous tient compagnie
f_{hyp_1}	de l' affection de l' amitié nous tient compagnie
f_{hyp_2}	de l' affection de l' amitié nous tient compagnie
$e_{hyp_{mt}}$	which gives us **the affection** , friendship and **keeps us airline** .
$e_{hyp_{slt1}}$	which **we affection of** friendship **we takes** company
$e_{hyp_{slt2}}$	which gives us **the affection of** friendship **we takes** company
e_{ref}	which gives us love , friendship and companionship .

Table 7. Example 3 - SLT hypothesis annotated with two methods - having the same number of *asr-errors*, but the different number of *slt-errors* extracted from *ASR1* and *ASR2* such as 2 B_ASR1, 2 B_ASR2, 12 B_MT, 14 B_SLT1, 9 B_SLT2.

led features could be found in (Besacier et al., 2014). For MT, we use a total of 24 major feature types which can be extracted with our word confidence estimation toolkit for MT (more details are given in (Servan et al., 2015)).

4.1 Experiments on 2-class Error Detection

Exp	MT+ASR feat. $p_{ASR}(q\|x_f,f)^{\alpha}$ $*p_{MT}(q\|e,f)^{1-\alpha}$	Joint feat. $p(q\|x_f,f,e)$
F-avg1 (ASR1)	58.07%	64.90%
F-avg2 (ASR2)	53.66%	64.17%

Table 8. Error Detection Performance (2-label) on SLT ouptut for *tst* set (training is made on *dev* set).

In this experiment, we evaluate the performance of our classifiers by using the average between the F-measure for $good$ labels and the F-measure for bad labels that are calculated by the common evaluation metrics: Precision, Recall and F-measure for $good/bad$ labels. Since two ASR systems are available, *F-avg1* is obtained for SLT based on $ASR1$ whereas *F-avg2* is obtained for SLT based on $ASR2$. The classifier is evaluated on the *tst* part of our corpus and trained on the *dev* part.

We report in Table 8 the baseline error detection results obtained using both MT and ASR features for a 2-class problem (error detection). More precisely we evaluate two different approaches (*combination* and *joint*):

- First system (MT+ASR feat.) combines the output of two separate classifiers based on ASR and MT features. In this approach, ASR-based confidence score of the source is projected to the target SLT output and combined with the MT-based confidence score as shown in Equation 2 (we did not tune the α coefficient and set it *a priori* to 0.5).
- Second system (joint feat.) trains a single error detection system for SLT (evaluating $p(q|x_f, f, e)$ as in Equation 1 using joint ASR and MT features. ASR features are projected to the target words using automatic word alignments.

Table 8 shows that joint ASR and MT features improve error detection performance over the use of simple combination (MT+ASR). Based on this result, only the joint approach is used in our 3-class experiments of next section. We also observe that F-measure decreases when ASR WER is lower (*F-avg2<F-avg1* while $WER_{ASR2} < WER_{ASR1}$). So error detection for SLT might be more complicated as ASR system improves.

These observations lead us to investigate the behaviour of our WCE approaches for a large range of $good/bad$ decision threshold.

While the previous tables provided WCE performance for a single point of interest ($good/bad$ decision threshold set to 0.5), the curves of Figure 2 show the full picture of our WCE systems (for SLT) using speech transcriptions systems $ASR1$ and $ASR2$, respectively. We observe that the classifier based on ASR features has a very different behaviour than the classifier based on MT features which explains why their simple combination (MT+ASR) does not work very well for the default decision threshold (0.5). However, for threshold above 0.75, the use of both ASR and MT features is slightly beneficial. This is interesting because higher thresholds improves the F-measure on bad labels (so improves error detection). Both curves are similar whatever the ASR system used. These results suggest that with enough development data for appropriate threshold tuning (which we do not have for this very new task), the use of both ASR and MT features should improve error detection in speech translation (blue and red curves are above the green curve for higher decision threshold [4]).

4.2 Experiments on 3-class Error Detection

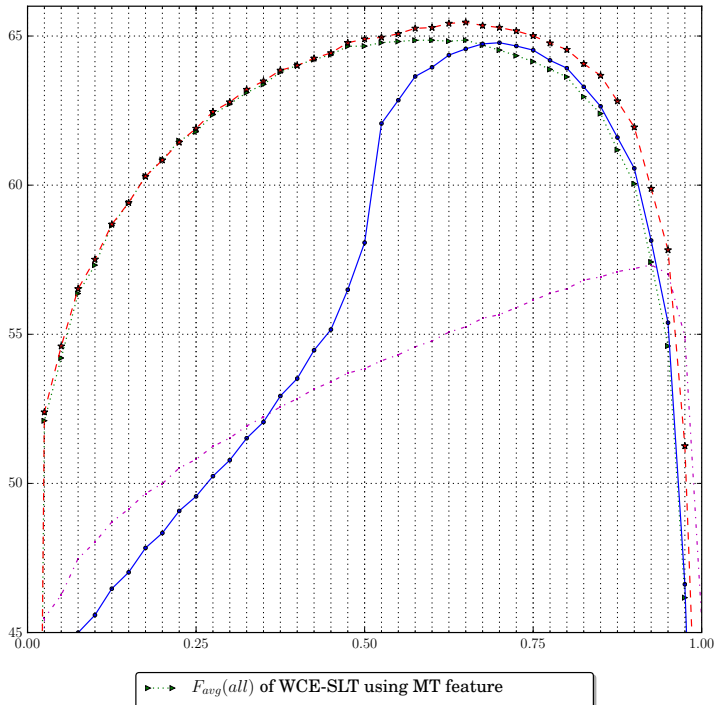
We report in Table 9 our first attempt to build an error detection system in SLT as a 3-class problem (*joint* approach only). We made our experiment by training and evaluating the model on *Intersection(m1, m2)* which corresponds to high confidence in the labels [5]. We compared two different approaches: *One-Step* is a single classifier for the 3-class problem while *Two-Step* first applies the 2 class (*G/B*) system and a second classifier distinguishes B_{ASR} and B_{MT} errors. Not much difference in F-measure is observed between both approaches. Table 10 also presents the confusion matrix between B_{ASR} and B_{MT} for the correctly detected (true) errors. Despite the relatively low F-scores of table 9, we see that our 3-labels classifier obtains encouraging confusion matrices in order to automatically disentangle B_{ASR} and B_{MT} on true errors.

5 Conclusions

This paper proposed to disentangle ASR and MT errors in speech translation. The binary error detection problem was recast as a 3-class labeling problem (*good, asr-error, mt-error*). First, two methods were proposed for the non trivial label setting and it was shown that both give

4. Corresponding to optimization of the F-measure on bad labels (errors).

5. However, we observed (results not reported here) that the use of different label sets (*Method 1*, *Method 2*, *Intersection(Method 1, Method 2)* does not have a strong influence on the results.



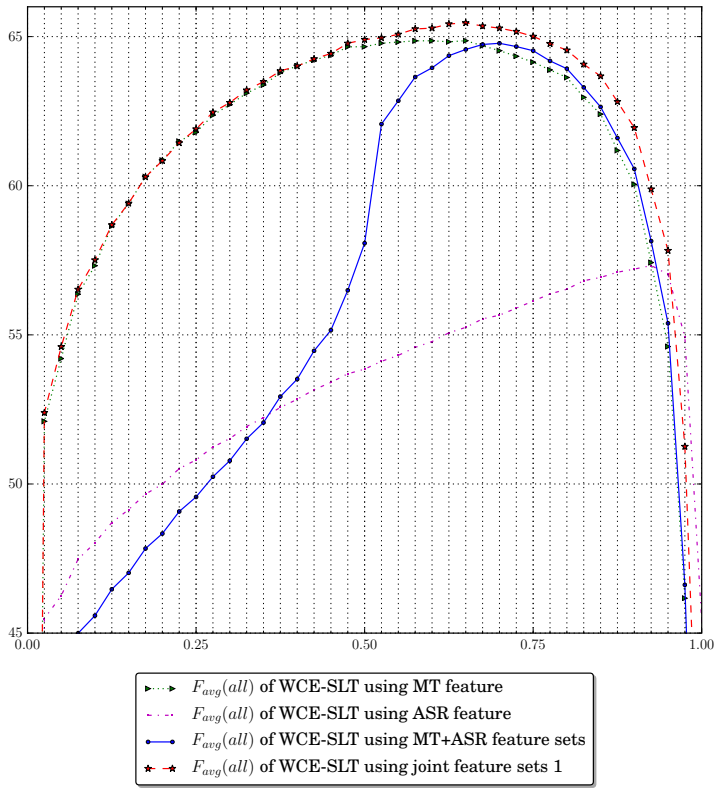


Figure 2. Evolution of system performance (y-axis - *F-mes1* - ASR1 and *F-mes2* - ASR2) for *tst* corpus (4050 utt) along decision threshold variation (x-axis) - training is made on *dev* corpus (2643 utt).

	2-class Full Corpus			3-class Intersection Corpus (m1, m2)			
				One-Step		Two-Step	
	ASR1	***ASR2***		***ASR1***	***ASR2***	***ASR1***	***ASR2***
F_G	81.79	83.17	F_G	85.00	85.00	84.00	85.00
F_B	48.00	45.17	F_{B_ASR}	44.00	42.00	44.00	42.00
			F_{B_MT}	14.00	15.00	16.00	17.00
F_{avg}	64.90	64.17	F_{avg}	47.67	47.33	48.00	48.00

Table 9. Error Detection Performance (2-label vs 3-label) on SLT output for tst set (training is made on *dev* set).

(1) Ref \ Hyp	***ASR1***		***ASR2***	
	B_ASR	***B_MT***	**B_ASR**	**B_MT**
B_ASR	*85.75%*	*14.25%*	81.57%	18.43%
B_MT	*44.46%*	*55.54%*	34.53%	65.47%
(2) Ref \ Hyp	***ASR1***		***ASR2***	
	B_ASR	***B_MT***	**B_ASR**	**B_MT**
B_ASR	*83.14%*	*16.86%*	80.02%	19.98%
B_MT	*49.41%*	*50.59%*	41.49%	58.51%

Table 10. Confusion Matrix on Correctly Detected Errors Subset for 3-class (1) One-Step; (2) Two-Step.

consistent results. Then, automatic detection of error types, using joint ASR and MT features, was evaluated and encouraging results were displayed on a French-English speech translation task. We believe that such a new task (not only detecting errors but also their cause) is interesting to build better informed speech translation systems, especially in interactive speech translation use cases.

Références

Besacier, L., Lecouteux, B., Luong, N. Q., Hour, K., and Hadjsalah, M. (2014). Word confidence estimation for speech translation. In *Proceedings of The International Workshop on Spoken Language Translation (IWSLT)*, Lake Tahoe, USA.

Besacier, L., Lecouteux, B., Luong, N.-Q., and Le, N.-T. (2015). Spoken language translation graphs re-decoding using automatic quality assessment. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Scotsdale, Arizona, United States.

Federico, M., Cettolo, M., Bentivogli, L., Paul, M., and Stüker, S. (2012). Overview of the IWSLT 2012 evaluation campaign. In *In proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*.

Galliano, S., Geoffrois, E., Gravier, G., Bonastre, J.-F., Mostefa, D., and Choukri, K. (2006). Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news. In *In Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC 2006)*, pages 315–320.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W.,

Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 177–180, Prague, Czech Republic.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting et labeling sequence data. In *Proceedings of ICML-01*, pages 282–289.

Lavergne, T., Cappé, O., and Yvon, F. (2010). Practical very large scale crfs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513.

Potet, M., Besacier, L., and Blanchon, H. (2010). The lig machine translation system for wmt 2010. In Workshop, A., editor, *Proceedings of the joint fifth Workshop on Statistical Machine Translation and Metrics MATR (WMT2010)*, Uppsala, Sweden.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Servan, C., Le, N.-T., Luong, N. Q., Lecouteux, B., and Besacier, L. (2015). An Open Source Toolkit for Word-level Confidence Estimation in Machine Translation. In *The 12th International Workshop on Spoken Language Translation (IWSLT'15)*, Da Nang, Vietnam.

Snover, M., Madnani, N., Dorr, B., and Schwartz, R. (2008). Terp system description. In *MetricsMATR workshop at AMTA*.

Temporality as Seen through Translation: A Case Study on Hindi Texts

Sabyasachi Kamila[†] sabysachi.pcs16@iitp.ac.in
Sukanta Sen[†] sukanta.pcs15@iitp.ac.in
Mohammad Hasanuzzaman[*] hasanuzzaman.im@gmail.com
Asif Ekbal[†] asif@iitp.ac.in
Andy Way[*] andy.way@adaptcentre.ie
Pushpak Bhattacharyya[†] pb@iitp.ac.in
[†]Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, India
[*]ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland

Abstract

Temporality has significantly contributed to various aspects of Natural Language Processing applications. In this paper, we determine the extent to which temporal orientation is preserved when a sentence is translated manually and automatically from the Hindi language to the English language. We show that the manually and automatically identified temporal orientation in English translated (both manual and automatic) sentences provides a good match with the temporal orientation of the Hindi texts. We also find that the task of manual temporal annotation becomes difficult in the translated texts while the automatic temporal processing system manages to correctly capture temporal information from the translations.

1 Introduction

There is a considerable academic and commercial interest in processing time information in text, where that information is expressed either explicitly, implicitly, or connotatively. Recognizing such information and exploiting it for Natural Language Processing (NLP) and Information Retrieval (IR) tasks are important features that can significantly improve the functionality of NLP/IR applications such as event timeline generation, question answering, and automatic summarization (Mani et al., 2005; Campos et al., 2014).

Earlier studies on temporal information processing have mainly focused on identifying temporal expressions fostered by TempEval challenges (Verhagen et al., 2010; UzZaman et al., 2013). More recently, new trends have emerged in the context of human temporal orientation, which refers to individual differences in the relative emphasis one places on the past, present, or future (Zimbardo and Boyd, 2015). Past studies have established consistent links between temporal orientation and demographic factors such as age, sex, gender, education, and psychological traits (Webley and Nyhus, 2006; Adams and Nettle, 2009; Schwartz et al., 2013; Zimbardo and Boyd, 2015). In order to create a measure of user-level human temporal orientation measure, a message-level[1] temporal

[1]Only the English message is considered from microblogs.

classifier of past, present, and future is used. For instance, the following microblog post "*can't wait to get a pint tonight*" is automatically tagged as *future* by the temporal classifier. Successful features include timexes, specific temporal (past, present, future) words from a commercial dictionary, but also *n*-grams.

Many tasks in NLP are language-dependent, i.e. the same approach cannot be applied across different languages. In this case, one naive way of temporality detection is to translate the text automatically into the desired language and then apply any temporality detector system. However, Machine Translation (MT) itself is a challenging task and often the meaning, sentiment (Salameh et al., 2015; Lohar et al., 2017), temporarily of a text may not be preserved in the target language.

In this paper, we discuss the degree of preservation of underlying temporal orientation of a sentence when it is translated from Hindi to English. We use Hindi and English temporality analysis systems (described in Section 6.2) as well as a state-of-the-art Hindi-to-English translation system (Koehn et al., 2003). From our experiments, we attempt to analyze all the possible cases and answer the following questions:

1. What is the accuracy of temporality prediction by an English temporality analysis system when *Hindi* texts are translated into *English*?

2. How good are these predictions when compared to the Hindi temporality system?

3. What is the loss in the temporality predictability when translating the Hindi text into English automatically vs. manually?

4. What is the difficulty level to determine temporality by humans in automatically translated texts from Hindi to English?

5. Which is better in detecting temporality of the Hindi text in the translated English text: (a) human temporal annotation of the translated text or (b) automatic temporality analysis of the translated text?

We know that linguistic divergences between a pair of languages play significant role while translating from one language to the other language, and hence it has a significant impact on the accuracy of an automatic computational model. Our specific goal here is to analyse the temporality predictability of the Hindi text after translation. However, we confer that similar experiments can be validated for other language pairs to determine the impact of translation on temporality.

We show the percentage of temporality preservation in the translated English sentences, with respect to the temporality of Hindi sentences. We also show that both manual and automatic translations produce a change of temporality from that of the Hindi texts; *past* and *present* sentences tends to be translated into sentences of *future* time. Our further analysis shows that some characteristics in the automatically translated text mislead humans to correctly detect the temporality of the source text, and some of those were correctly classified by the automatic temporal analysis system.

Our contributions can be summarized as follows: i). to the best of our knowledge this is the first systematic attempt which presents a study whether temporality is preserved after translation; ii). we prepare a benchmark setup by creating three annotated datasets- Hindi texts, manual and automatic translated English texts labeled with three temporal classes, namely *past*, *present* and *future*; and iii). detecting the change of temporality in both manually a automatically translated sentences.

2 Related Works

Temporality has recently received increased attention in NLP and IR. The introduction of the TempEval task (Verhagen et al., 2009) and subsequent challenges (TempEval-2 and -3) in the Semantic Evaluation workshop series have clearly established the importance of time in dealing with different NLP tasks.

According to Metzger (2007), time is one of the key five aspects that determines a document credibility besides relevance, accuracy, objectivity and coverage. Given this, the value of information or its quality is intrinsically time-dependent. As a consequence, a new research field called Temporal Information Retrieval (T-IR) has emerged and deals with all classical IR tasks such as crawling (Kulkarni et al., 2011), indexing (Anand et al., 2012) or ranking (Kanhabua et al., 2011) from the viewpoint of time. From an application perspective of T-IR, Campos et al. (2014) proposed a solution for temporal classification of queries by identifying the top relevant dates in web snippets with respect to a given implicit temporal query, with temporal disambiguation performed through a distributional metric called GTE. Competitions like the NTCIR-11 Temporalia task (Joho et al., 2014) further pushed this idea and proposed to distinguish whether a given query is related to *past*, *recency*, *future* or *atemporal*. In order to push forward further research in temporal NLP and IR, Dias et al. (2014) developed TempoWordNet (TWn), an extension of WordNet (Miller, 1995), where each synset is augmented with its temporal connotation (past, present, future, or atemporal). Same kind of approach was followed for Hindi to create a lexical resource, namely TempoHindiWordNet (Pawar et al., 2016).

At the same time, there has been quite a few works on MT involving the Hindi-English language pair. Most of these systems aim to translate from English to Hindi or Indian languages (Dave et al., 2001; Sinha and Jain, 2003; Sinha and Thakur, 2005; Ananthakrishnan et al., 2006; Dungarwal et al., 2014; Sachdeva et al., 2014; Sen et al., 2016). One of the major challenges in MT between Hindi to English is the syntactic divergence. English follows the word order of Subject-Verb-Object (SVO) whereas Hindi follows Subject-Object-Verb (SOV). Ramanathan et al. (2008) have shown that simple syntactic transformation of the English language to meet the syntax of Hindi can improve translation quality. For our Hindi-English translation system, we follow the standard phrase based statistical MT (Koehn et al., 2003) approach.

3 Methodology Overview

We present our experimental setup to study the impact of translation on temporality, as follows:

1. Collect a Hindi dataset (*Hi*) described in Section 4.2.

2. Manually translate *Hi* into English (*En*). We refer to these English translations as *En(Manl.Trans.)*.

3. Automatically translate *Hi* into *En*. We refer to these English translations as *En(Auto.Trans.)*.

4. Manually annotate *Hi* for temporality. We call these *Hi(Manl.Tempo.)*.

5. Manually annotate all English datasets (*En(Manl.Trans.)* and *En(Auto.Trans.)*) for temporality. We call those *En(Manl.Trans., Manl.Tempo.)* and *En(Auto.Trans., Manl.Tempo)*, respectively.

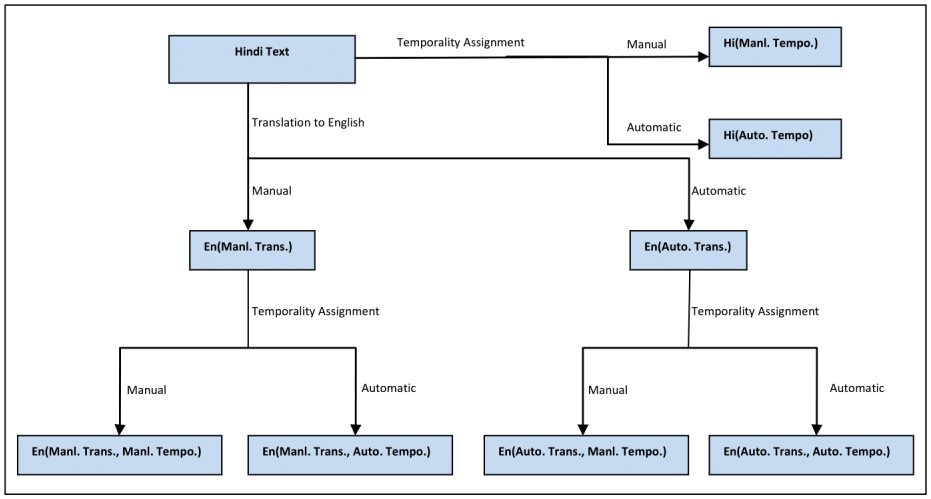


Figure 1: Proposed Architecture.

6. Run a Hindi temporality detector on *Hi*, creating *Hi(Auto. Tempo.)*

7. Run an English temporality detector on all the English datasets (*En(Manl. Trans.)* and *En(Auto. Trans.)*) creating *En(Manl. Trans., Auto. Tempo.)* and *En(Auto. Trans., Auto. Tempo.)*, respectively.

8. The procedural steps are depicted in Figure 1.

After creating various temporality-labeled datasets, we can compare the pairs of datasets to draw inferences. For example, comparison of the labels for *En(Manl. Trans., Manl. Tempo.)* and *En(Auto. Trans., Manl. Tempo.)* will show how the automatic translation affects the manual temporal levels with respect to the manual translation. The comparison will also show, for example, the extent to which a *past* sentence tends to be translated as a *present* sentence. The comparison of the dataset pairs (*Hi(Manl. Tempo.)* vs. *En(Auto. Trans., Auto. Tempo.)*) will show whether the idea of first translating a Hindi sentence into English and then using the automatic temporality detection is feasible or not. Section 5 demonstrates the procedure of Hindi to English translation. Section 6 describes the ways of finding temporality for different datasets i.e. *Hi*, *En(Manl. Trans.)* and *En(Auto. Trans.)*, both manually and automatically. Finally, Section 7 discusses the temporal error rate and analysis of different test cases.

4 Dataset

For our experiments, we use a parallel corpus of Hindi-English created in Bojar et al. (2014). This corpus contains 274k Hindi-English parallel sentences. The training and test sets for temporal tagging are described in Section 4.1 and 4.2. For MT, the details of training, test and development sets are mentioned in Section 5.

4.1 Training Set

We select past-, present-, and future-oriented texts using a manually selected high precision list of 50 seed terms.These are terms that capture temporal dimensions of texts with very few false positives, though the recall of these terms is low. In order to increase

the recall, and to capture new terms that are good examples of past, present, and future, we expand our initial seed terms using a query expansion technique. For English, we use the publicly available word2vec[2] vectors that are trained on Google News corpus. For Hindi, we employ a continuously distributed vector representation of words using the continuous Skip-gram model (also known as Word2Vec) proposed by Mikolov et al. (2013) and trained on a corpus of around 44 million Hindi sentences developed by Bojar et al. (2014) with dimension set to 300 and window size set to 7.

Given the vector representations for the terms, we calculate the similarity scores between the pairs of terms in our vocabulary using cosine similarity. The top-10 similar terms for each seed term are selected for the expansion of the initial seed list. We again filter the whole collection of texts using the newly added seed terms. Table 1 shows few examples of expanded terms for some of the initial seed terms. There are some unrelated keywords in the expanded seed list due to the automatic process of keyword selection.

	Temporality	**Initial Seeds**	**Expanded Seed Terms**
Hindi	Past	गत (gata-past)[3]	विगत (vigata-last/past), पिछले (piChale-last/previous), बीते (bIte-past/bygone), पिछल्ले (piChalle-last/previous), विगत (vigata-last/past), गतवर्ष (gatavarSha-last year)
	Present	फ़िलहाल (phailahAla-at the moment)	फ़िलहाल (phailahAla-at the moment), अभी (abhI-now), अब (aba-now), फिलवक्त (philavakta-philanthropy), बहरहाल (baharahAla-nevertheless), ख़ैर (khaaira-well), हाल-फिलहाल (hAla-philahAla-most recently)
	Future	वादा (vAdA-promise)	वादे (vAde-promises), वायदा (vAyadA-futures), ऐलान (ailAna-announce), एलान (elAna-announce), दावा (dAvA-claim), आग्रह (Agraha-request)
English	Past	yesterday	yesterday, Earlier, Last, Shortly_afterwards, Meanwhile
	Present	currently	presently, Currently, now, currenty, still, already, iscurrently, actively
	Future	promise	promises, pledge, vow, commitment, hope, expect, vowing

Table 1: Examples of initial seed terms and their expanded terms.

Following this procedure, we create datasets for both Hindi and English containing 40K sentences each. Finally, we create our training set of 15k for both Hindi and English separately,[4] which consists of equally distributed past, present and future sentences. For the similar reason justified in Schwartz et al. (2015), we only considered past, present and future categories. Some example sentences are:

- नासा ने कल्पना के नाम से एक सुपर कंप्यूटर समर्पित किया है *(nAsA ne kalpanA ke nAma se eka supara kaMpyUTara samarpita kiyA hai-NASA has dedicated a super computer in the name of Kalpana)*, ***past***.

- अब ये अपने दोस्तों को बुलाने लगा है *(aba ye apane dostoM ko bulAne lagA hai-Now he is calling his friends)*, ***present***.

[2] `https://code.google.com/p/word2vec/`

[3] Henceforth, all the Hindi examples are represented by Hindi texts, ITRANS representations and using equivalent English translations.

[4] As our aim is to check whether temporality changes after translation or not, we are not using the translated version of the Hindi to create the training set for English.

- मेरे फूल को क्षण-भर में नष्ट हो जाने का जोखिम है *(mere phUla ko kShaNa-bhara meM naShTa ho jAne kA jokhima hai-My flower is at risk of being destroyed momentarily)*, ***future***.

4.2 Test Set

At first, we manually annotate the Hindi sentences with appropriate temporal categories from the same Hindi-English Bojar corpus. We made it sure that no training instances are being included. Finally, we select 996 sentences of past, present and future temporal classes. We call these 996 Hindi sentences *Hi* and the manually tagged *Hi* as *Hi(Manl. Tempo.)*. We then consider the manually translated English sentences(*En(Manl. Trans.)*) from *Hi* and then manually annotate them for temporality. We call these *En(Manl. Trans., Manl. Tempo.)*. We then manually annotate the automatically translated English sentences (*En(Auto. Trans.)*) from *Hi* for temporality. We call them as *En(Auto. Trans., Manl. Tempo.)*. Finally, we obtain three temporality tagged test sets, namely *Hi(Manl. Tempo.)*, *En(Manl. Trans., Manl. Tempo.)* and *En(Auto. Trans., Manl. Tempo.)*. We use *Hi* as the test set in Section 5 for MT.

5 Translation of Hindi to English

Our Hindi-English translation system, a phrase-based statistical MT system (Koehn et al., 2003), was built using Hindi-English parallel Bojar corpus (Bojar et al., 2014). We first remove the set (*Hi*) described in Section 4.2 from the corpus which is used as the test set for our MT system. We thereafter randomly select training and development sets from the rest of the corpus.

Set	#Sentences	#Tokens	
		En	Hi
Train	260,711	2,993,765	3,281,273
Test	996	23,806	27,012
Development	1000	12,480	14,153

Table 2: Statistics of data sets used in Hindi-English MT system

We tokenize, true-case and remove longer sentences as part of the preprocessing of the data. English sentences are tokenized using the *tokenizer.perl*[5] script, and we used the Indic_NLP_Library[6] for tokenizing Hindi sentences. After preprocessing, the training and development sets contain 260,711 and 1,000 parallel sentences, respectively. Details of the data sets are shown in Table 2.

For training, we use the Moses (Koehn et al., 2007) SMT system. We use KenLM (Heafield, 2011) for building a 4-gram language model and GIZA++ (Och and Ney, 2003) with the grow-diag-final-and heuristic for extracting phrases from the parallel corpus. The trained system is tuned using Minimum Error Rate Training (Och, 2003). For other parameters of Moses, default values are used. Automatic evaluation of our translation system achieves a BLEU (Papineni et al., 2002) score of 16.66.

6 Temporal Tagging of Sentences

We detect temporality in one Hindi dataset (*Hi*) and two English datasets *En(Manl. Trans.)*, *En(Auto. Trans.)* which denote manual and automatic translations from Hindi

[5] `https://github.com/moses-smt/mosesdecoder/blob/RELEASE-3.0/scripts/tokenizer/tokenizer.perl`

[6] `https://bitbucket.org/anoopk/indic\_nlp\_library`

to English language, respectively, as described in Section 4. We deploy both manual as well as automatic methods for temporal tagging.

6.1 Manual Temporal Tagging of Sentences

We create the datasets following manual annotation process as described in Section 4.2. Three annotators were asked to annotate based on the time sense in the sentences using past, present and future temporal categories. For the Hindi dataset(*Hi*), we considered only the temporal sentences, namely past, present and future. While annotating the two English datasets (*En(Manl. Trans.)*, *En(Auto. Trans.)*), we consider another category, namely *atemporal* apart from the three temporal categories. The reason for this consideration was to verify our hypothesis as to whether temporality is lost after translation. Finally, we consider sentences based on majority voting. We did not stick to the tense-based tagging as it sometimes misled the annotators to detect the actual temporality of the sentence. For example, consider the following sentence:

- आगामी छुट्टियों के लिए मेरे पास एक अच्छी योजना है *(AgAmI ChuTTiyoM ke lie mere pAsa eka achChI yojanA hai-I have a nice plan for the upcoming holidays).*

Here the tense of the verb *"have"* is *present* while the time sense of the sentence refers to *"future"*. Annotations also vary from person to person as any concrete definition of words does not exist; rather it is defined by the context appearing in the sentence. Finally, we obtain three sets of manually annotated datasets, namely *Hi(Manl. Tempo.)*, *En(Manl. Trans., Manl. Tempo.)* and *En(Auto. Trans., Manl. Tempo.)*. The temporality statistics are depicted in Table 3.

Datasets	**Temporality(%)**		
	Past	**Present**	**Future**
Hi(Manl. Tempo.)	32.83	24.80	42.37
En(Manl. Trans., Manl. Tempo.)	38.95	19.58	32.93
En(Auto. Trans., Manl. Tempo.)	41.15	11.75	34.74

Table 3: Class distribution of the manually annotated temporal datasets

From the statistics in Table 3, we can see that even after manual translation, loss of temporality is possible. The amount of loss in temporality in the dataset *En(Manl. Trans., Manl. Tempo.)* is 8.54%. In the automatically translated dataset *En(Auto. Trans., Manl. Tempo.)* the amount of loss in temporality is 13.35%, which is more than that of the manually translated set. Examples of these two cases are as follows:

1. **Manual Translation:** The temporality of the Hindi sentence *"*मानसिक रोग संबंधित लक्ष्य *(mAnasika roga saMbaMdhita lakShya)"* is *future*, but in the manually translated sentence *"Mental illness targets"*, the annotators tag it as *atemporal*. We observe that in the manually translated set, the temporality loss is mainly due to the incorrect temporal annotation rather than the incorrect manual translation. One of the possible reasons may be that the annotators were instructed not to see the temporal class of the Hindi sentence while labeling the English side. This was done to reduce bias.

2. **MT:** The Hindi sentence *"*ग्रामीण चीन में आर्थिक नवीनीकरण हुये हैं *(grAmINa chIna meM Arthika navInIkaraNa huye haiM- Economic Renewal happened in Rural China)"*, which has temporality *past*. This sentence is automatically translated as *"in rural areas are bound to China"* which becomes a factual text with no temporal sense. From our observation, we can say that the loss of temporality, in this

case, is mostly because of the wrong automatic translation rather than the the wrong manual annotation.

6.2 Automatic Temporal Classifier

We use a supervised machine learning-based approach for automatic sentence-level temporal tagging. For this experiment, we use the training set and test set as described in Section 4. We automatically classify three datasets, namely *Hi*, *En(Manl. Trans.)*, and *En(Auto. Trans.)*, for temporality in one of the three temporal categories, namely past, present or future. We employ one-vs.-rest approach for both our generation models as well as for evaluation. Our test set construction follows the same approach. For classification, we use Support Vector Machine (Joachims, 2002) classifier with word-unigram as a feature. Classification yields three sets of temporal datasets, named as *Hi(Auto. Tempo.)*, *En(Manl. Trans., Auto. Tempo.)* and *En(Auto. Trans., Auto. Tempo.)*. The class distribution of these temporal datasets is shown in Table 4.

Datasets	**Temporality(%)**		
	Past	**Present**	**Future**
Hi(Auto. Tempo.)	32.96	30.53	36.51
En(Manl. Trans., Auto. Tempo.)	16.12	20.97	62.91
En(Auto. Trans., Auto. Tempo.)	19.56	13.15	67.28

Table 4: Class distribution of the automatically tagged temporal datasets

7 Temporality after Translation

We generate all the manually and automatically labeled datasets mentioned in the experimental setup in Section 3 using the methods and systems described in Sections 3, 5 and 6. Results of class distribution in Table 3 can be compared with that in Table 4. The comparison of temporality labels between different data pairs is depicted in Table 5.

Data Pair	**Match(%)**
a. Hi(Manl. Tempo.) - Hi(Auto. Tempo.)	72.39
b. Hi(Manl. Tempo.) -En(Manl. Trans., Manl. Tempo.)	67.47
c. Hi(Manl. Tempo.) - En(Manl. Trans., Auto. Tempo.)	66.42
d. Hi(Manl. Tempo.) - En(Auto. Trans., Manl. Tempo.)	59.33
e. Hi(Manl. Tempo.) - En(Auto. Trans., Auto. Tempo.)	62.49
f. En(Manl. Trans., Manl. Tempo.) - En(Auto. Trans., Manl. Tempo.)	62.35
g. En(Manl. Trans., Manl. Tempo.) - En(Manl. Trans., Auto. Tempo.)	69.59
h. En(Auto. Trans., Manl. Tempo.) - En(Auto. Trans., Auto. Tempo.)	69.17

Table 5: Percentage of matching between pairs of temporality labeled datasets.

Row a., in Table 5 shows that the match percentage between the manual temporality and automatic temporality of Hindi texts is 72.39% which is the accuracy of the automatic temporality analysis system for Hindi.

Row b. shows the percentage match between the two manually temporal tagged datasets (*Hi(Manl.Tempo.)* and *En(Manl. Trans., Manl.Tempo.)*). We observe that two labels match only 67.47% of the time. It shows that the English translation does affect temporality.

Row c. shows the temporality match between the automatic temporality on manually translated texts and *Hi(Manl.Tempo.)*. Observe that the match for this pair is

66.42%, which is not too much lower than 67.47% obtained in the case of manual temporal tagging. This shows that English temporal system performs rather well. More importantly, the English automatic temporality analysis on the automatically translated texts shows a match of 62.49% (row e.), which makes this choice feasible for the temporality analysis of non-English texts.

Rows d. and e. show the temporality match of Hindi manual temporality with manual and automatic temporal labeling of the automatically translated texts, respectively. As the translation is automatic here, we expect these match percentages to be lower than those in rows b. and c. where the translation is manual, and the results show the same. However, we unexpectedly find the number for row e. to be higher than that of row d. This shows that some characteristics of the automatically translated text mislead humans with regards to the true temporality of the source text. However, this claim needs further insight in future.

Row f. shows the match between the manual temporal labels of manual and automatic translated English texts which is only 62.35%. Row g. shows the accuracy of the English automatic temporal analysis system when the translation is manual. The result of 69.59% shows that the quality of the English temporal analysis system is good, irrespective of human errors.

Row h. shows the accuracy of the English automatic temporal analysis system when the translation is automatic. In this case, the system's accuracy of 69.17% again shows that MT greatly impacts temporality.

We manually examine several Machine translated texts to understand the reason for incorrect annotations by humans with respect to Hindi annotation (row d. of Table 5). Most cases were due to translation errors where the temporal words were either lost or replaced by the other temporal words. Table 6 shows some examples of possible error cases. We observe that often the linking verb changes to a linking verb of a different temporality. In some cases, due to the change in the structure of the sentence, the temporality changes. Temporality loss happens mainly for the loss of action words and it occurs for all types of temporal sentences (past, present and future).

MT Error	**Temporality**
Change of linking verb after translation: **Hindi text:** तब लोगों को और अधिक बदला लेने की संभावना थी (taba logoM ko aura adhika badalA lene kI saMbhAvanA thI - Then people were more likely to take revenge.) **MT output:** when people are more likely to take revenge.	 past future
Change of linking verb after translation: **Hindi text:** मैं अपनी नियति का पीछा कर रहा हूं (maiM apanI niyati kA pIChA kara rahA hUM - I'm following my destiny.) **MT output:** I was in pursuit of his destiny.	 present past
Structural Change after translation: **Hindi text:** हमें उनकी प्रगति दिखाईये (hameM unakI pragati dikhAIye - Show us their progress.) **MT output:** their progress shows us.	 future present
Loss of action word: **Hindi text:** लेकिन सचाई शायद कुछ और निकले (lekina sachAI shAyada kuCha aura nikale - But maybe a different truth can come out) **MT output:** but the truth and perhaps some.	 future atemporal

Table 6: Examples of temporality change or loss due to different types of MT errors.

We analyze two cases to understand whether automatic temporality detection can be effective over the manual temporality in the translated instances. Our first case is comprised of the results in row b. and row c. of Table 5, where the translation is manual. There are some instances where the automatic temporality on the manually translated text correctly tags texts, while the manual temporality fails. The reason behind this is that the system can learn an appropriate model even from the mistranslated text. For example, consider the following case:

- **Hindi text:** "कि अगर किसी ने माना कि यह एक खतरा नहीं है *(ki agara kisI ne mAnA ki yaha eka khatarA nahIM hai)".*
- **Correct English translation:** *"That if somebody believes that this is not a threat".*
- **Manual English translation:** *"That if anybody believes that it wasn't such a threat".*

In the example, the temporal tag for the Hindi sentence is *future*, but when it is manually translated, the tag becomes *past*. In this case, the automatic English temporal tagger correctly predicts it as *future*. We observe that there are 6.7% instances in the manually translated English texts which are manually tagged incorrectly with respect to the Hindi text's temporality but correctly tagged by the automatic English temporal tagger.

Our second case is based on the results in row d. and row e. in Table 5, where the translation is automatic. The automatic temporal analysis system correctly tags several automatically translated instances (where manual labeling fails) for the same reason as for the first case. Consider the following examples:

- **Hindi text:** "तीसरी योजना में लगभग सभी अतिरिक्त क्षमता सार्वजनिक क्षेत्र को देते हुए इस्पात पिंडों का लक्ष्य 102 लाख टन पर निश्चित किया गया *(tIsarI yojanA meM lagabhaga sabhI atirikata kShamatA sArvajanika kShetra ko dete hue ispAta piMDoM kA lakShya 102 lAkha Tana para nishchita kiyA gayA)".*
- **Correct English translation:** *"Giving almost all the additional capacities to the public sector in the third plan, the goal of steel bodies was fixed at 102 lakh tonnes".*
- **MT output:** *"in the Third Plan the public sector almost all the additional capacity to steel ingots target of 102 million tonnes".*

In this case, the original temporal class in Hindi is *past*. In the machine translated English text, human experts annotate it as *future*, but the automatic temporal tagger tags it correctly as *past*. This case is quite interesting as despite obtaining some ungrammatical and unstructured sentences using machine translation, the automatic temporal tagger still correctly predicts temporality for some sentences. Our analysis shows that 8.14% temporal instances appear in the automatically translated English texts which are manually tagged incorrectly with respect to Hindi texts but correctly tagged by the automatic English temporal tagger.

8 Conclusion

In this paper, we present a case study on how machine translation affects temporality when the text is translated from Hindi to English. To the best of our knowledge this is the first study that systematically analyses various aspects of temporality preservation after translation. We create benchmark setups by creating manually labeled datasets for various test case scenarios. Our thorough investigation shows that temporality can

be both lost and altered while text is translated from one source to the other target language. We also observe that the accuracy of the automatic temporal tagger in the automatically translated texts produces competitive results with respect to the accuracy of the automatic temporal tagger in the manually translated texts.

In future, we will explore these possible cases and further determine whether temporality preservation can improve the translation quality or not. We also propose to extend our study to more language pairs and use neural MT system for translation.

Acknowledgments

This research is partially supported by ADAPT Centre for Digital Content Technology, funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Adams, J. and Nettle, D. (2009). Time Perspective, Personality and Smoking, Body Mass, and Physical Activity: An Empirical Study. *British journal of health psychology*, 14(1):83–105.

Anand, A., Bedathur, S. J., Berberich, K., and Schenkel, R. (2012). Index Maintenance for Time-travel Text Search. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval*, pages 235–244, Portland, OR, USA.

Ananthakrishnan, R., Kavitha, M., Jayprasad, J. H., Shekhar, R., and Bade, S. (2006). MaTra: A Practical Approach to Fully-automatic Indicative English-Hindi Machine Translation. In *Symposium on Modeling and Shallow Parsing of Indian Languages*, pages 1–8, Mumbai, India.

Bojar, O., Diatka, V., Rychlý, P., Stranák, P., Suchomel, V., Tamchyna, A., and Zeman, D. (2014). HindEnCorp - Hindi-English and Hindi-only Corpus for Machine Translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 3550–3555, Reykjavik, Iceland.

Campos, R., Dias, G., Jorge, A. M., and Jatowt, A. (2014). Survey of Temporal Information Retrieval and Related Applications. *ACM Comput. Surv.*, 47(2):15:1–15:41.

Dave, S., Parikh, J., and Bhattacharyya, P. (2001). Interlingua-based English–Hindi Machine Translation and Language Divergence. *Machine Translation*, 16(4):251–304.

Dias, G. H., Hasanuzzaman, M., Ferrari, S., and Mathet, Y. (2014). TempoWordNet for Sentence Time Tagging. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, pages 833–838, Geneva, Switzerland.

Dungarwal, P., Chatterjee, R., Mishra, A., Kunchukuttan, A., Shah, R., and Bhattacharyya, P. (2014). The IIT Bombay Hindi-English Translation System at WMT 2014. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 90–96, Baltimore, Maryland, USA.

Heafield, K. (2011). KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland,.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms.* Kluwer Academic Publisher, Dordrecht, Netherlands.

Joho, H., Jatowt, A., Blanco, R., Naka, H., and Yamamoto, S. (2014). Overview of NTCIR-11 Temporal Information Access (Temporalia) Task. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies*, pages 429–437, Tokyo, Japan.

Kanhabua, N., Blanco, R., and Matthews, M. (2011). Ranking Related News Predictions. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 755–764, Beijing, China.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180, Prague, Czech Republic.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54, Edmonton, Canada.

Kulkarni, A., Teevan, J., Svore, K. M., and Dumais, S. T. (2011). Understanding Temporal Query Dynamics. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining*, pages 167–176, Hong Kong, China.

Lohar, P., Afli, H., and Way, A. (2017). Maintaining Sentiment Polarity in Translation of User-Generated Content. *The Prague Bulletin of Mathematical Linguistics*, 108(1):73–84.

Mani, I., Pustejovsky, J., and Gaizauskas, R. (2005). *The Language of Time: a Reader*, volume 126. Oxford University Press, Oxford, UK.

Metzger, M. (2007). Making Sense of Credibility on the Web: Models for Evaluating Online Information and Recommendations for Future Research. *Journal of the American Society for Information Science and Technology*, 58(13):2078–2091.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in neural information processing systems*, pages 3111–3119, Lake Tahoe, Nevada, United States.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.

Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167, Sapporo, Japan.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Carious Statistical Alignment Models. *Computational linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, Philadelphia, Pennsylvania.

Pawar, D., Hasanuzzaman, M., and Ekbal, A. (2016). Building Tempo-HindiWordNet: A Resource for Effective Temporal Information Access in Hindi. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016*, pages 3752–3759, Portorož, Slovenia.

Ramanathan, A., Hegde, J., Shah, R. M., Bhattacharyya, P., and Sasikumar, M. (2008). Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation. In *Third International Joint Conference on Natural Language Processing*, pages 513–520, Hyderabad, India.

Sachdeva, K., Srivastava, R., Jain, S., and Sharma, D. M. (2014). Hindi to English Machine Translation: Using Effective Selection in Multi-Model SMT. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 1807–1811, Reykjavik, Iceland.

Salameh, M., Mohammad, S., and Kiritchenko, S. (2015). Sentiment after Translation: A Case-Study on Arabic Social Media Posts. In *Proceedings of The 2015 Conference of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies (NAACL)*, pages 767–777, Denver, Colorado, USA.

Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., Shah, A., Kosinski, M., Stillwell, D., Seligman, M. E., et al. (2013). Personality, Gender, and Age in the Language of Social Media: The Open-Vocabulary Approach. *PloS one*, 8(9):1–16.

Schwartz, H. A., Park, G., Sap, M., Weingarten, E., Eichstaedt, J., Kern, M., Berger, J., Seligman, M., and Ungar, L. (2015). Extracting Human Temporal Orientation in Facebook Language. In *Proceedings of The 2015 Conference of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies (NAACL)*, pages 409–419, Denver, Colorado, USA.

Sen, S., Banik, D., Ekbal, A., and Bhattacharyya, P. (2016). IITP English-Hindi Machine Translation System at WAT 2016. In *Proceedings of the 3rd Workshop on Asian Translation (WAT)*, pages 216–222, Osaka, Japan.

Sinha, R. and Jain, A. (2003). AnglaHindi: an English to Hindi Machine-aided Translation System. In *MT Summit IX*, pages 494–497, New Orleans, USA.

Sinha, R. and Thakur, A. (2005). Machine Translation of Bi-lingual Hindi-English (Hinglish) Text. In *10th Machine Translation summit (MT Summit X)*, pages 149–156, Phuket, Thailand.

UzZaman, N., Llorens, H., Derczynski, L., Allen, J. F., Verhagen, M., and Pustejovsky, J. (2013). SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013*, pages 1–9, Atlanta, Georgia, USA.

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Moszkowicz, J., and Pustejovsky, J. (2009). The TempEval Challenge: Identifying Temporal Relations in Text. *Language Resources and Evaluation*, 43(2):161–179.

Verhagen, M., Saurí, R., Caselli, T., and Pustejovsky, J. (2010). SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010*, pages 57–62, Uppsala, Sweden.

Webley, P. and Nyhus, E. K. (2006). Parents' Influence on Children's Future Orientation and Saving. *Journal of Economic Psychology*, 27(1):140–164.

Zimbardo, P. G. and Boyd, J. N. (2015). Putting Time in Perspective: A Valid, Reliable Individual-Differences Metric. In *Time Perspective Theory; Review, Research and Application*, pages 17–55. Springer, Berlin, Germany.

A Neural Network Transliteration Model in Low Resource Settings

Tan Le le.ngoc_tan@courrier.uqam.ca
Universite du Quebec a Montreal, Canada

Fatiha Sadat sadat.fatiha@uqam.ca
Universite du Quebec a Montreal, Canada

Abstract

Transliteration is the process of converting a text in one script to another, guided by phonetic clues. This conversion requires an important set of rules defined by expert linguists to determine how the phonemes are aligned and to take into account the phonology system of the target language. The problem with under-resourced language pairs remains the lack of linguistic resources. In this research, we present a recurrent neural network based approach to overcome the transliteration problem for a low-resource language pair, with an application on the French-Vietnamese language pair. Our system requires a small bilingual learning dataset. We obtained promising results with a large gain of BLEU-score and a reduction in translation errors rate (TER) and phonemes errors rate (PER), compared to other systems.

1 Introduction

Transliteration consists of a process of transforming a word from a writing system (called *source word*) to a phonetically equivalent word of another writing system (called *target word*) (Knight and Graehl, 1998). Many of the named entities (*i.e. person names, location, organization, technical terms, etc.*) are often transliterated from a source language to a target language when translation is difficult or impossible. Transliteration can be considered as a sub-task of machine translation (MT).

Named entities constitute an open morphological class. Person names and organizations names, which are never seen before in the learning phase, often appear in the new documents. It is critical that MT systems address this issue. Integrating a transliteration module within a MT system remains a solution for solving out-of-vocabulary words (OOV) having the type of named entities.

Moreover, with the evolution of high technologies and the globalization of commerce, people tend to invent new words. It is very difficult to define all the possible rules of phonetic transformation between the source language and the target language.

In this research, we propose a method of low resource machine transliteration using recurrent neural network (RNN) based model. This task automatically predicts the phonemic representation of a word in the target language given a new word in the source language that does not exist in the dictionary of bilingual phonetics. We are interested in solving out-of-vocabulary words considered as proper names or technical terms from a machine translation system for a under-resourced language pair, with application for French-Vietnamese.

Our contribution is to show how, with a small bilingual learning dataset, we can train a RNN-based model for low resource machine transliteration. To the problem of sparse data due

to the low resource languages, we apply an algorithm to re-rank the list of k-best results from the baseline transliteration model.

The structure of the article is as follows: Section 2 presents the state of the art on transliteration. Section 3 describes our proposed approach. Section 4, we present our experiments and compare the performance of our system with other systems as well as errors analysis. Finally, in section 5, we conclude with some perspectives.

2 Related work

Since 2009, various transliteration systems have been proposed during the Named Entities Workshop evaluation campaigns [1] (Duan et al., 2016). These campaigns consist of transliterating from English into languages with a wide variety of writing systems, including Hindi, Tamil, Russian, Kannada, Chinese, Korean, Thai and Japanese. We can see that the romanization of non-Latin writing systems remains a complex computational task that is highly dependent on a language.

Through this workshop, much progress has been made in the methodologies with an emergence of different approaches, such as grapheme in the phoneme (Finch and Sumita, 2010; Ngo et al., 2015), based on statistics like automatic translation (Laurent et al., 2009; Nicolai et al., 2015) as well as neural networks (Finch et al., 2015, 2016; Shao and Nivre, 2016; Thu et al., 2016).

The variety of writing systems adds another important challenge in the extraction of named entities and automatic transliteration. All these difficulties are aggravated by the lack of bilingual dictionaries of proper names, ambiguities of transcription as well as orthographic variation in a language.

(Lo et al., 2016) used a semi-supervised transliteration model built on a seed corpus mined from the standard parallel training data, in order to improve the Russian-English machine translation system for WMT 2016.

(Ngo et al., 2015) proposed a statistical model for a language pair with English-Vietnamese language, with a phonological constraint on the syllables. Their system has achieved better performance than the base system, based on rules, with a 70% reduction in error rates.

(Cao et al., 2010) also applied the statistical-based approach as automatic translation in the transliteration task for a language pair with little English-Vietnamese language, with a performance of 63% of BLEU (Papineni et al., 2002). Our proposed approach is totally different, except for the same preparation of the bilingual phonetic dictionary learning. We propose a step of rescoring k-best results from the baseline transliteration system to solve the problem of scattered data due to the low resource language.

3 Proposed Approach

3.1 Phonology of Vietnamese

The structure of syllables in French is very rich, with a variety of structures such as CV, CVC, $CCVCC$, etc. Where C is a consonant and V is a vowel. On the other hand, the structure of syllables in Vietnamese is very simple. One of the linguistic peculiarities of Vietnamese is that a word consists of a syllable or several syllables (Phe, 1997). A syllable in Vietnamese is constituted with the following structure:

$$Syllable = Onset + Vowel + Coda$$

The boundary of a syllable depends on consonant groups (*onset and coda*) and vowels. The Vietnamese has a Latin alphabet with 29 letters. There are 12 vowels and 17 consonants unigrams, 9 consonants bi-grams and 1 tri-gram. The vowels are $V = \{$*"a", "ă", "â", "e", "ê",*

[1] http://workshop.colips.org/news2016/

"i", "o", "ô", "ơ", "u", "ư", "y" }. The consonants are $Onset = \{$*"b", "ch", "c", "d", "đ", "gi", "gh", "g", "h", "kh", "k", "l", "m", "ngh", "ng", "nh", "n", "ph", "q", "r", "s", "th", "tr", "t", "v", "x", "p"*$\}$. Among these consonants, there are 8 in tail $Coda = \{$*"c", "ch", "n", "nh", "ng", "m", "p", "t"*$\}$. The Vietnamese has 6 lexical tones such as *up (i.e. có = own), broken (i.e. mỹ = american), flat (i.e. ba = father), interrogative (i.e. thuỷ = water), down (i.e. trà = tea) and low (i.e. lại = coming)*. (Phe, 1997) found about 10,000 syllables for Vietnamese. In this research work, we focus only on the grapheme and the phoneme of all words in the bilingual dictionary.

3.2 Multi-joint sequence model

The approach of graphemes-to-phonemes with a multi-joint sequence model has been proposed by (Deligne et al., 1995). This is one of the most popular approaches in the task of converting graphemes into phonemes by machine learning. The main idea consists in generating both the sequences at the level of graphemes and at the level of phonemes by means of a single joined sequence of the linguistic units which represent all the symbols of graphemes and phonemes. In fact, the aim of this approach is to find a sequence of phonemes Y defined by $Y = P_1^m = \{p_1, p_2, ..., p_m\}$, Given a sequence of graphemes X defined by $X = G_1^n = \{g_1, g_2, ..., g_n\}$. The problem can become the estimation of the most optimal Y phoneme sequence, which maximizes their conditional probability as in the following equation 1:

$$\hat{e} = argmax_{i \in Y}\ p(Y|X) \tag{1}$$

According to the Bayes' Theorem:

$$\hat{e} = argmax_{i \in Y} \frac{p(X|Y)\ p(Y)}{p(X)} \tag{2}$$

Because $p(X)$ is independent of all the phoneme sequences Y, the equation 2 can be simplified as follows:

$$\hat{e} = argmax_{i \in Y}\ p(X|Y)\ p(Y) \tag{3}$$

3.3 Recurrent neural network based sequence model for small data

Figure 1 shows the architecture of a RNN model adapted from (Yao and Zweig, 2015). A RNN model takes an input of a sequence of vectors $(x_1, x_2, ..., x_n)$ and produces an output of a sequence of vectors $(h_1, h_2, ..., h_n)$ to represent the information at each input step. LSTMs (*Long-Short Term Memory*), which are a type of RNNs, have been designed to incorporate a memory cell which can protect and control the cell state. They use several gates to control the amount of information from the previous states which should be forgotten and the information from the inputs which should be updated to the memory cell (Hochreiter and Schmidhuber, 1997). The formulas that govern the computation happening in a RNN are as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{4}$$

$$c_t = (1 - i_t) \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{5}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{6}$$

$$h_t = o_t \odot tanh(c_t) \tag{7}$$

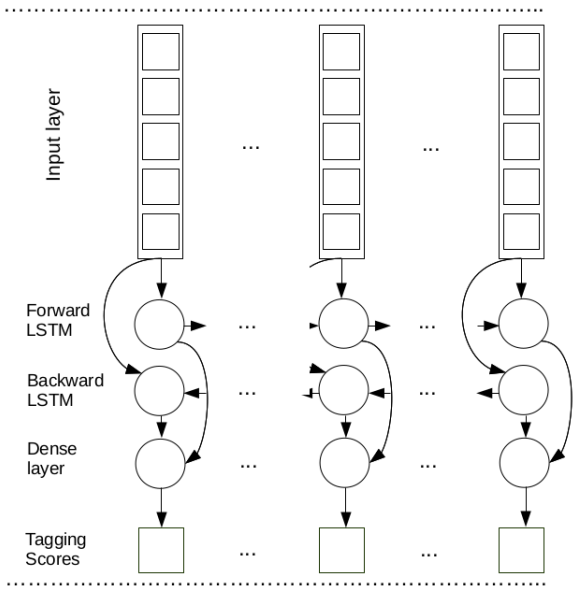


Figure 1: Our recurrent neural network based model architecture

where σ is the element-wise sigmoid function, and $\odot$ is the element-wise product. c_t and o_t are the cell state and the output at the step *t*, respectively.

There are many variants of LSTM implementations. LSTM sequence-to-sequence models were successfully applied in various tasks, including machine translation (Sutskever et al., 2014) and grapheme-to-phoneme (Yao and Zweig, 2015).

Our approach consists of three main steps: *(1) pre-processing, (2) creating a RNN-based model, and (3) re-ranking the k-best*. The whole process is illustrated in Figure 1.

(1) First, we collect bilingual phonetic linguistic resources for a low resource language pair, here French-Vietnamese. Then, this learning data is pre-processed with normalization in miniscule as well as a segmentation of syllables in Vietnamese, which is explained in section 3.1 - phonology of Vietnamese.

(2) Then we train a RNN-based model.

(3) Finally, we implement an additional module to re-rank the list of k-best results from the transliteration model of bilingual proper names. Inspired from (Bhargava et al., 2011), we use the algorithm of *Support Vector Machines (SVM)* for this module, with different characteristics such as *phonemic alignment scores, orthographic and phonetic similarities and length difference between each pair of graphemes-phonemes*.

4 Experimentation

4.1 Data preparation

We use a bilingual phonetic dictionary that has been collected from the news websites as presented in (Cao et al., 2010). The data learning has 4,259 pairs of bilingual French-Vietnamese proper names, with a set of vocabularies that contains 31 graphemes in the French source side, and 71 phonemes in the Vietnamese target side. We find that most of the bilingual proper names are person names, location names and organization names. To overcome the problem of

the scattering of learning data, we perform the pre-processing with normalization for the entire data (Figure 2).

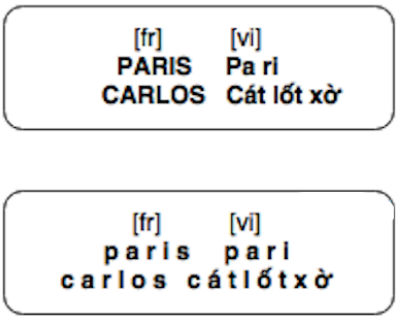


Figure 2: Illustration of the bilingual phonetic dictionary and the pre-processing results

Inspired by the phrases-based statistical approach (*pbSMT*), we consider a baseline system by applying this approach, but based on characters. We implement a *pbSMT* system with the *Moses*[2] (Koehn et al., 2007). We use *mGIZA* (Gao and Vogel, 2008) to align the corpus to the character level, and *SRILM* (Stolcke et al., 2002) to create a 5-gram language model for Vietnamese. While the *pbSMT* systems implemented by (Finch and Sumita, 2010)(Nicolai et al., 2015) have not taken into account word reordering, we will test various word reordering models offered by *Moses*.

We apply *Sequitur-G2P* [3] tool to train our transliteration model of bilingual proper names for the French-Vietnamese language pair.

We used 2-layer bi-directional *Long Short-Term Memory* (LSTM) cells (Hochreiter and Schmidhuber, 1997) for the RNN-based model, with a 64-dimensional projection layer to encode the input sequences and 64 nodes in each hidden layer. We used the *'sgd'* (*Stochastic Gradient Descent*) optimizer to learn the weights of the network with a learning rate of 0.5. We used *g2p-sep2seq*[4] toolkit. This implementation is based on python *TensorFlow*, which allows an efficient training on both CPU and GPU.

We implement a *SVM re-classification module* using the *LinearSVC* library of *scikit-learn*[5] for the purpose of rescoring the best hypotheses from a list of k-best (with $k = 100$) results obtained by the baseline transliteration system.

4.2 Evaluation

The bilingual phonetic dictionary of learning is split into one training set, one development set and one test set with a ratio of 80 %, 10% and 10 % respectively.

We apply different evaluation metrics such as *BiLingual Evaluation Understudy* (Papineni et al., 2002), *Translation Error Rate (TER)* (Snover et al., 2009) with a tool of *multeval* version 0.5.1 [6] (Clark et al., 2011).

For the phonemic error rate, we use a *Phoneme Error Rate (PER)* metric with *SCLITE* (a tool for calculating scores and evaluating the results of speech recognition systems) NIST

[2] http://www.statmt.org/moses/
[3] https://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html
[4] https://github.com/cmusphinx/g2p-seq2seq
[5] http://scikit-learn.org/stable/modules/Generated/sklearn.svm.LinearSVC.html
[6] http://www.cs.cmu.edu/~jhclark/downloads/multeval-0.5.1.tgz

SCTK version 2.4.10[7]. The method of calculating the error rate of phonemes with *SCLITE* is similar to that for words (*Word Error Rate*). We use the Levenshtein distance measure in this work. This distance measure is shown in the equation 8, where N is the number of phonemes, as follows:

$$PER = \frac{\sum_{i=1}^{n} d_{edit}(hypothese_i, reference_i)}{|N|} \quad (8)$$

In order to evaluate our proposed approach, we implement three systems, including the baseline system (*pbSMT*), system 1 (*Sequitur-g2p*) and system 2 (*our proposed approach*) (Table 1).

If we compare the baseline system with system 1, the difference in their performance is minor. System 1 seems slightly more efficient than the baseline system, with a gain of +4.40% of BLEU, as well as reductions of -4.30% and -6.20% of translation errors (TER) and phonemes (PER) respectively.

On the other hand, by comparing the baseline system and the system 2 (*our proposed approach*), we note significant results with a gain of +26.95% of BLEU, reductions in translation errors (TER) and phonemes (PER) with -9.30% and -31.30% respectively.

In addition, the performance of system 2 is higher than that of system 1, with a gain of +22.55% of BLEU, reductions of -5.0% and -25.10% of translation error (TER) and phonemic (PER) rates respectively (Table 1).

In general, the proposed approach has achieved very well the transliteration task with the significant gains and can reduce the phoneme error rate. We can observe the output quality of the proposed approach, which is based on the recurrent neural network, is more fluid, coherent and with fewer errors than the baseline and the system 1, which are both based on the statistical approach. We carry out an error analysis on the next section to more details.

Metric	**System**	**Average**	$\overline{s}_{sel}$	s_{Test}	p**-value**
BLEU ↑	Baseline (*pbSMT*)	61.30	1.70	-	-
	System 1 (*Sequitur-g2p*)	65.70	1.70	-	0.79
	System 2 (*our approach*)	**88.25**	1.50	-	0.01
TER ↓	Baseline (*pbSMT*)	24.80	1.20	-	-
	System 1 (*Sequitur-g2p*)	20.50	1.20	-	0.13
	System 2 (*our approach*)	**15.50**	1.00	-	0.00
PER ↓	Baseline (*pbSMT*)	44.20	-	-	-
	System 1 (*Sequitur-g2p*)	38.00	-	-	-
	System 2 (*our approach*)	**12.90**	-	-	-

Table 1: Evaluation about scoring for all systems : **BLEU**, **TER** and **PER**.
p **-values** are relative to the base system and indicate whether a difference of this magnitude between the baseline system and other systems. $\overline{s}_{sel}$ indicates the variance due to the selection of the test.

4.3 Error analysis

We perform an error analysis in the three evaluation systems to better understand the errors likely to predicted phonemes from French to Vietnamese.

First, we check the top-5 best results, for example, from our transliteration model before re-ranking the list of k-best results (Tables 2 and 3). We find that the first result of transliteration in Vietnamese, having the best probability given a grapheme in French, is not always the correct

[7] http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm

PARIS			MANHATTAN		
No	**TOP-5**	**Probability**	**No**	**TOP-5**	**Probability**
1	**p a r i**	**0.633242**	**1**	m a n h á t t â n	0.321082
2	p a r í t	0.153536	**2**	**m a n h á t t a n**	**0.288677**
3	b a r i	0,065151	**3**	m â n h á t t â n	0.080221
4	b a r í t	0.037314	**4**	m â n h á t t a n	0.072125
5	p a r í t x ơ	0.028526	**5**	m a h á t t â n	0.058193

Table 2: Illustration of the transliteration predictions of the named entities obtained by our proposed approach before the re-ranking of the list of k-best results, with the top-5 ($k = 5$) first best results for the named entities : *PARIS* and *MANHATTAN*

GAULOISE		
No	**TOP-5**	**Probability**
1	g ô l o a x ơ	0.102710
2	g ô n l o a x ơ	0.096937
3	**g ô l o a d ờ**	**0.092091**
4	g ô n l o i d ờ	0.086915
5	g ô l o a ờ	0.072750

Table 3: Illustration of the transliteration predictions of the named entities obtained by our proposed approach before the re-ranking of the list of k-best results, with the top-5 ($k = 5$) first best results for the named entities : *GAULOISE*

transliteration. Therefore, it is essential to re-classify the best hypotheses among a list of k-best (with $k = 100$) results of the transliteration system. For example: *PARIS -> p a r i (p = 0.633242), MANHATTAN -> m a n h á t t a n (p = 0.288677) and GAULOISE -> g ô l o a d ờ (p = 0.092091).*

Evaluation	Proper Names	IPA format	Hypothesis	Reference
Baseline	paris	**p a r i**	p a r í t	p a r i
	tigrane	**t i g r a n e**	t i g ờ r a n nơ	t i g ờ r a n nơ
	toulouse	**t u l u s e**	t u l u x ơ	t u l u gi ơ
	tours	**t u r**	t u ố c t x ơ	t u a
	truffaut	**t r y f o**	tr uy ph ố t	tr uy ph ô
	zurich	**z y r i k**	gi uy r i	gi uy r í ch
System 1	paris	**p a r i**	p a r í t	p a r i
	tigrane	**t i g r a n e**	t i g ờ r a n nơ	t i g ờ r a n nơ
	toulouse	**t u l u s e**	t u l u x ơ	t u l u gi ơ
	tours	**t u r**	t u ố c t x ơ	t u a
	truffaut	**t r y f o**	tr uy ph ố t	tr uy ph ô
	zurich	**z y r i k**	gi uy r i	gi uy r í ch
System 2	paris	**p a r i**	p a r i	p a r i
	tigrane	**t i g r a n e**	t i g ờ r a n nơ	t i g ờ r a n nơ
	toulouse	**t u l u s e**	t u l u x ơ	t u l u gi ơ
	tours	**t u r**	t u a	t u a
	truffaut	**t r y f o**	tr uy ph ô	tr uy ph ô
	zurich	**z y r i k**	gi uy r í ch	gi uy r í ch

Table 4: Examples of transliteration prediction results by all systems, with IPA (*International Phonetic Alphabet*) format as *ground truth*, hypothesis and reference for six proper names such as *PARIS, TIGRANE, TOULOUSE, TOURS, TRUFFAUT and ZURICH*

We then perform a comparison of the transliteration prediction results of the named entities between the three evaluation systems with some named entities that are not yet seen during the learning phase (Table 4). We find that the baseline system (*pbSMT*) and system 1 (*Sequitur-G2P*) have incorrectly transliterated the proper names such as PARIS, TOURS, TRUFFAUT and ZURICH, while system 2 (*our proposed approach*) provided good results. We note that the three systems encounter difficulties in predicting optimally all the possibilities of transliteration of bilingual proper names due to the original variety of named entities (*i.e. French, English, Italian, Russian, etc.*) As well as the pronunciation of different tail syllables such as *"-er" (/e/ = ê or /ϵ/ = e), "-s" (xơ or ϕ) , "-te" (tơ or ϕ) or "-x" (ích or ϕ).*

5 Conclusion

In this paper, we presented a recurrent neural network based approach to overcome the transliteration problem for a low resource language pair, with an application on the French-Vietnamese language pair. Results show that the RNN-based model outperforms both the phrasal MT and the *Sequitur-G2P* baselines. The RNN-based model yields significant improvements in error rates over state-of-the-art systems.

To our knowledge, we are not aware of any research nor study that analyzes Vietnamese in the transliteration task. Our research focusing on machine transliteration is the first work for the French-Vietnamese bilingual low resource language pair. This system requires only a bilingual phonetic dictionary. This system has the capacity to learn, automatically, the linguistic regularities from this bilingual phonetic dataset.

In future work, we intend to develop our approach with a larger bilingual phonetic dataset as well as to study other approaches such as *attentional mechanism*, in order to improve the performance of neural network models when low amounts of training data are available.

References

Bhargava, A., Hauer, B., and Kondrak, G. (2011). Leveraging transliterations from multiple languages. In *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*, pages 36–40.

Cao, N. X., Pham, N. M., and Vu, Q. H. (2010). Comparative analysis of transliteration techniques based on statistical machine translation and joint-sequence model. In *Proceedings of the 2010 Symposium on Information and Communication Technology*, pages 59–63. Association for Computing Machinery.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 176–181. Association for Computational Linguistics.

Deligne, S., Yvon, F., and Bimbot, F. (1995). Variable-length sequence matching for phonetic transcription using joint multigrams. In *Fourth European Conference on Speech Communication and Technology*.

Duan, X., Banchs, R. E., Zhang, M., Li, H., and Kumaran, A. (2016). Report of news 2016 machine transliteration shared task. *Association for Computational Linguistics 2016*, pages 58–72.

Finch, A., Liu, L., Wang, X., and Sumita, E. (2015). Neural network transduction models in transliteration generation. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, page 61.

Finch, A., Liu, L., Wang, X., and Sumita, E. (2016). Target-bidirectional neural models for machine transliteration. *Association for Computational Linguistics 2016*, pages 78–82.

Finch, A. and Sumita, E. (2010). Transliteration using a phrase-based statistical machine translation system to re-score the output of a joint multigram model. In *Proceedings of the 2010 Named Entities Workshop*, pages 48–52. Association for Computational Linguistics.

Gao, Q. and Vogel, S. (2008). Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Knight, K. and Graehl, J. (1998). Machine transliteration. *Computational Linguistics*, 24(4):599–612.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Laurent, A., Deléglise, P., Meignier, S., and Spécinov-Trélazé, F. (2009). Grapheme to phoneme conversion using an smt system. In *Proceedings of INTERSPEECH, ISCA*, pages 708–711.

Lo, C.-k., Cherry, C., Foster, G., Stewart, D., Islam, R., Kazantseva, A., and Kuhn, R. (2016). Nrc russian-english machine translation system for wmt 2016. In *Proceedings of the First Conference on Machine Translation, Berlin, Germany. Association for Computational Linguistics*.

Ngo, H. G., Chen, N. F., Nguyen, B. M., Ma, B., and Li, H. (2015). Phonology-augmented statistical transliteration for low-resource languages. In *Interspeech*, pages 3670–3674.

Nicolai, G., Hauer, B., Salameh, M., St Arnaud, A., Xu, Y., Yao, L., and Kondrak, G. (2015). Multiple system combination for transliteration. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, pages 72–79.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Phe, H. (1997). Vietnamese dictionary. *Vietnam Lexicography Centre, Da Nang Publishing House*.

Shao, Y. and Nivre, J. (2016). Applying neural networks to english-chinese named entitytransliteration. In *Sixth Named Entity Workshop, joint with 54th Association for Computational Linguistics*.

Snover, M. G., Madnani, N., Dorr, B., and Schwartz, R. (2009). Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23(2-3):117–127.

Stolcke, A. et al. (2002). Srilm-an extensible language modeling toolkit. In *Interspeech*, volume 2002, page 2002.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Thu, Y. K., Pa, W. P., Sagisaka, Y., and Iwahashi, N. (2016). Comparison of grapheme–to–phoneme conversion methods on a myanmar pronunciation dictionary. *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing 2016*, pages 11–22.

Yao, K. and Zweig, G. (2015). Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1506.00196*.