SYSTRAN Purely Neural MT Engines for WMT2017

Yongchao Deng, Jungi Kim, Guillaume Klein, Catherine Kobus, Natalia Segal, Christophe Servan, Bo Wang, Dakun Zhang, Josep Crego, Jean Senellart

firstname.lastname@systrangroup.com
SYSTRAN / 5 rue Feydeau, 75002 Paris, France

Abstract

This paper describes SYSTRAN's systems submitted to the WMT 2017 shared news translation task for English-German, in both translation directions. Our systems are built using OpenNMT[1], an open-source neural machine translation system, implementing sequence-to-sequence models with LSTM encoder/decoders and attention. We experimented using monolingual data automatically back-translated. Our resulting models are further hyper-specialised with an adaptation technique that finely tunes models according to the evaluation test sentences.

1 Introduction

We participated in the WMT 2017 shared news translation task on two different translation directions: English→German and German→English.

The paper is structured as follows: Section 2 overviews our neural MT engine. Section 3 describes the set of experiments carried out to build the English→German and German→English neural translation models. Experiments and results are detailed in Section 3. Finally, conclusions are drawn in Section 4.

2 Neural MT System

Neural machine translation (NMT) is a new methodology for machine translation that has led to remarkable improvements, particularly in terms of human evaluation, compared to rule-based and statistical machine translation (SMT) systems (Crego et al., 2016; Wu et al., 2016). NMT has now become a widely-applied technique for machine translation, as well as an effective approach for other related NLP tasks such as dialogue, parsing, and summarisation.

Our NMT system (Klein et al., 2017) follows the architecture presented in (Bahdanau et al., 2014). It is implemented as an encoder-decoder network with multiple layers of a RNN with Long Short-Term Memory (LSTM) hidden units (Zaremba et al., 2014). Figure 1 illustrates an schematic view of the MT network.

Source words are first mapped to word vectors and then fed into a bidirectional recurrent neural network (RNN) that reads an input sequence $s = (s_1, ..., s_J)$. Upon seeing the <eos> symbol, the final time step initialises a target RNN. The decoder is a RNN that predicts a target sequence $t = (t_1, ..., t_I)$, being J and I respectively the source and target sentence lengths. Translation is finished when the decoder predicts the <eos> symbol.

The left-hand side of the figure illustrates the bidirectional encoder, which actually consists of two independent LSTM encoders: one encoding the normal sequence (solid lines) that calculates a forward sequence of hidden states $(\overrightarrow{h_1}, ..., \overrightarrow{h_J})$, the second encoder reads the input sequence in reversed order (dotted lines) and calculates the backward sequence $(\overleftarrow{h_1}, ..., \overleftarrow{h_J})$. The final encoder outputs $(\overline{h}_1, ..., \overline{h}_J)$ consist of the sum of both encoders final outputs. The right-hand side of the figure illustrates the RNN decoder. Each word t_i is predicted based on a recurrent hidden state h_i and a context vector c_i that aims at capturing relevant source-side information.

Figure 2 illustrates the attention layer. It implements the "general" attentional architecture from (Luong et al., 2015). The idea of a global attentional model is to consider all the hidden states of the encoder when deriving the context vector c_t. Hence, global alignment weights a_t are derived by

[1] http://opennmt.net

Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 265–270
Copenhagen, Denmark, September 711, 2017.

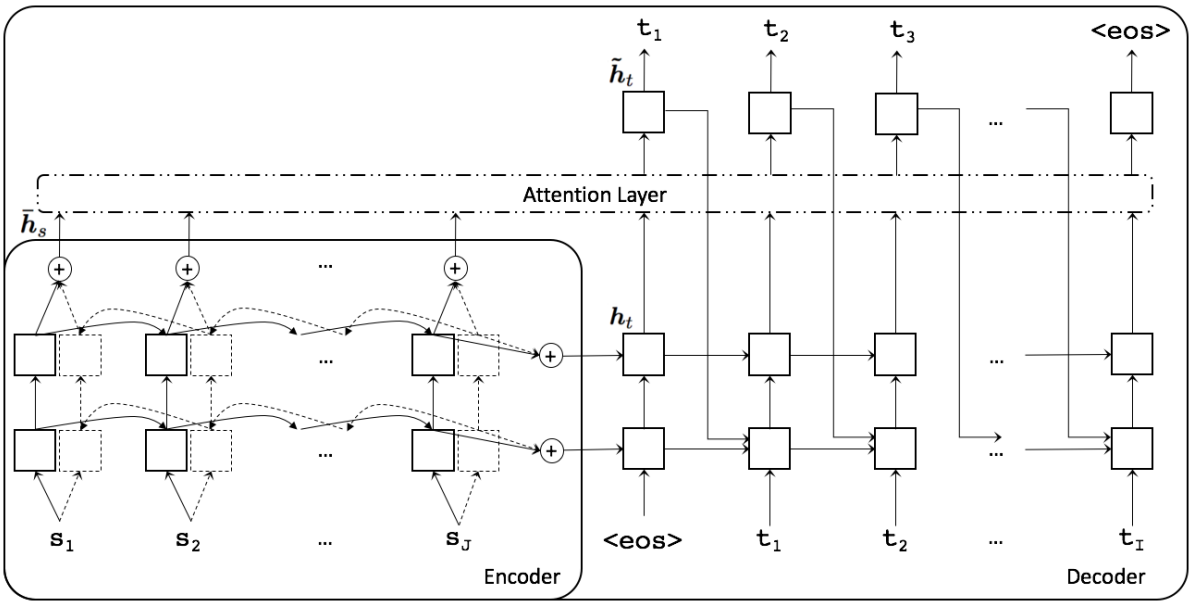

Figure 1: Schematic view of our MT network.

comparing the current target hidden state h_t with each source hidden state $\overline{h}_s$:

$$a_t(s) = \frac{exp(score(h_t, \overline{h}_s))}{\sum_{s'} exp(score(h_t, \overline{h}_{s'}))}$$

with the content-based score function:

$$score(h_t, \overline{h}_s) = h_t^T W_a \overline{h}_s$$

Given the alignment vector as weights, the context vector c_t is computed as the weighted average over all the source hidden states.

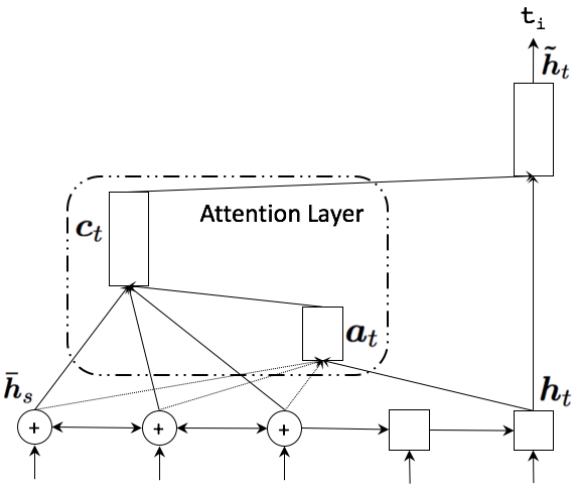

Figure 2: Attention layer of the MT network.

Note that for the sake of simplicity figure 1 illustrates a two-layers LSTM encoder/decoder while any arbitrary number of LSTM layers can be stacked. More details about our system can be found in (Crego et al., 2016).

3 Experiments

In this section we detail the corpora and training experiments used to build our English↔German neural translation models.

3.1 Corpora

We used the parallel corpora made available for the shared task: *Europarl v7*, *Common Crawl corpus*, *News Commentary v12* and *Rapid corpus of EU press releases*. Both English and German texts were preprocessed with standard tokenisation tools. German words were further preprocessed to split compounds, following a similar algorithm as the built-in for Moses. Additional monolingual data was also used for both German and English available for the shared task: *News Crawl: articles from 2016*. Basic statistics of the tokenised data are available in Table 1.

We used a byte pair encoding technique[2] (BPE) to segment word forms and achieve open-vocabulary translation with a fixed vocabulary of 30,000 source and target tokens. BPE was originally devised as a compression algorithm, adapted to word segmentation (Sennrich et al., 2016b). It recursively replaces frequent consecutive bytes with a symbol that does not occur elsewhere. Each

[2] https://github.com/rsennrich/subword-nmt

	#sents	#words	vocab.	L_{mean}
Parallel				
En	4.6M	103.7M	627k	22.6
De	4.6M	104.5M	836k	22.8
Monolingual				
En	20,6M	463,6M	1.18M	22.5
De	34,7M	620,8M	3.36M	17.8

Table 1: *English-German parallel and monolingual corpus statistics. L_{mean} indicates mean sentence lengths. M stand for millions, k for thousands.*

such replacement is called a merge, and the number of merges is a tuneable parameter. Encodings were computed over the union of both German and English training corpora after preprocessing, aiming at improving consistency between source and target segmentations.

Finally, case information was considered by the network as an additional feature. It allowed us to work with a lowercased vocabulary and treat recasing as a separate problem (Crego et al., 2016).

3.2 Training Details

All experiments employ the NMT system detailed in Section 2. The encoder and the decoder consist of a four-layer stacked LSTM with $1,000$ cells each. We use a bidirectional RNN encoder. Size of word embedding is 500 cells. We use stochastic gradient descent, a minibatch size of 64 sentences and 0.3 for dropout probability. Maximum sentence length is set to 80 tokens. All experiments are performed on NVidia GeForce GTX 1080 on a single GPU per optimisation work. Newstest2008 (2008) is employed as validation test set and newstest from 2009 to 2016 (2009-16) as internal test sets.

3.2.1 Training on parallel data

Table 2 outlines training work. All parallel data (**P**) is used on each training epoch. Row LR indicates the learning rate value used for each epoch. Note that learning rate was initially set to 1.0 during several epochs until no or little perplexity (PPL) reduction is measured on the validation set. Afterwards, additional epochs are performed with learning rate decayed by 0.7 at each epoch. BLEU score (averaged over the eight internal test sets) after each training epoch is also shown. Note that all BLEU scores shown in this paper are computed using `multi-bleu.perl`[3]. Training time per epoch is also shown in row Time measured in number of hours.

As expected, a perplexity reduction is observed for the initial epochs, until epochs 9 (German→English) and 8 (English→German) where little or no improvement is observed. The decay mode is then started allowing to further boost accuracy (between 1.5 and 2.0 BLEU points) after 5 additional epochs.

3.2.2 Training on parallel and synthetic data

Following (Sennrich et al., 2016a), we selected a subset of the available target-side in-domain monolingual corpora, translate it into the source side (back-translate) of the respective language pair, and then use this synthetic parallel data for training. The best performing models for each translation direction (epoch 13 on Table 2 of both translation directions) were used to back-translate monolingual data. (Sennrich et al., 2016a) motivate the use of monolingual data with domain adaptation, reducing overfitting, and better modelling of fluency.

Synthetic corpus was then divided into i different splits containing each 4.5 million sentence pairs (except for the last split that contains less sentences). Table 3 shows continuation of the training work using at each epoch the union of the entire parallel data together with a split of the monolingual back-translated data (**P+M**$_i$). Hence, balancing the amount of reference and synthetic data, summing up to around 9 million sentence pairs per epoch. Note that training work described in Table 3 is built as continuation of the model at epoch 13 on Table 2. Table 3 shows also BLEU scores over newstest2017 for the best performing network.

As for the experiments detailed in Table 2, once all splits of the synthetic corpus were used to train our models with learning rate always set to 1.0 (5 epochs for German→English and 8 epochs for English→German), we began a decay mode. In this case, we decided to reduce the amount of training examples from 9 to 5 millions due to time restrictions. To select the training data we employed the algorithm detailed in (Moore and Lewis, 2010). It aims at identifying sentences in a generic corpus that are closer to domain-

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

Epoch	1	2	3	4	5	6	7	8	9	10	11	12	13
German→English													
Data	P	P	P	P	P	P	P	P	P	P	P	P	P
Time (hours)	24	24	24	24	24	24	24	24	24	24	24	24	24
LR	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7^1	0.7^2	0.7^3	0.7^4
PPL (2008)	20.90	17.01	15.38	14.67	14.18	13.75	13.57	13.29	13.00	12.47	12.05	11.49	11.40
BLEU (2009-16)	20.07	22.06	23.02	24.17	24.59	24.40	24.99	25.11	25.42	25.65	26.14	26.48	26.87
English→German													
Data	P	P	P	P	P	P	P	P	P	P	P	P	P
Time (hours)	24	24	24	24	24	24	24	24	24	24	24	24	24
LR	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7^1	0.7^2	0.7^3	0.7^4	0.7^5
PPL (2008)	20.85	16.52	14.84	13.89	13.62	13.13	12.59	12.66	11.72	11.20	10.94	10.75	10.55
BLEU (2009-16)	15.63	17.41	18.85	19.61	19.92	20.38	20.34	20.55	21.13	21.63	21.70	22.22	22.50

Table 2: *Training on parallel data.*

Epoch	1	2	3	4	5	6	7	8	9	10	11	12	13
German→English													
Data	P+M$_1$	P+M$_2$	P+M$_3$	P+M$_4$	P+M$_5$	P'+M'	P'+M'	P'+M'	P'+M'	P'+M'			
Time (hours)	45	45	45	45	32	25	25	25	25	25			
LR	1.0	1.0	1.0	1.0	1.0	0.7^1	0.7^2	0.7^3	0.7^4	0.7^5			
PPL (2008)	13.33	13.23	13.26	1347	12.63	12.25	11.87	11.60	11.40	11.33			
BLEU (2009-16)	26.85	27.37	27.37	27.01	27.77	27.91	28.34	28.54	28.75	28.73			
BLEU (2017)										32.35			
English→German													
Data	P+M$_1$	P+M$_2$	P+M$_3$	P+M$_4$	P+M$_5$	P+M$_6$	P+M$_7$	P+M$_8$	P'+M'	P'+M'	P'+M'	P'+M'	P'+M'
Time (hours)	46	46	46	46	46	46	46	40	25	25	25	25	25
LR	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7^1	0.7^2	0.7^3	0.7^4	0.7^5
PPL (2008)	12.87	12.91	12.38	12.23	12.19	12.00	12.26	11.65	11.51	11.19	10.80	10.70	10.58
BLEU (2009-16)	21.81	22.26	22.52	22.65	22.59	22.75	22.79	22.93	23.35	23.56	23.79	23.96	24.07
BLEU (2017)													26.41

Table 3: *Training on parallel and synthetic data.*

specific data. Figure 3 outlines the algorithm. In our experiments, parallel and monolingual back-translated corpus are considered as the generic corpora (**P+M**) while all available newstest test sets, from 2009 to 2017, are considered as the domain-specific data (**T**). Hence, we aim at selecting from **P+M** the closest 5 million sentences to the newstest2009-17 data (2.5 from the **P** and 2.5 from the **M** subsets).

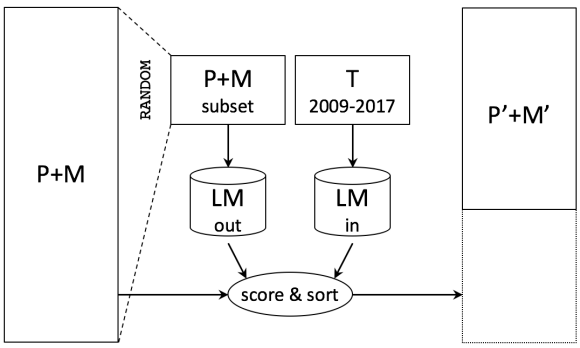

Figure 3: Data selection process.

Obviously, we base our selection procedure on the source-side text of each translation direction as references for newstest2017 are not available.

Sentences s of the generic corpus are scored in terms of cross-entropy computed from two language models: a 3-gram LM trained on the domain-specific data $H_{in}(s)$ and a 3-gram LM trained on a random sample taken from itself $H_{out}(s)$. Finally, sentences of the generic corpus are sorted regarding the computation of the difference between domain-specific and generic scores $H_{in}(s) - H_{out}(s)$ (score & sort).

3.2.3 Hyper-specialisation on news test sets

Similar to domain adaptation, we explore a post-process approach, which hyper-specialises a neural network to a specific domain by running additional training epochs over newly available in-domain data (Servan et al., 2016). In our context, we utilise all newstest sets (**T**) (around $25,000$ sentences), as in-domain data and run a single learning iteration in order to fine tune the resulting network. Translations are not available for newstest2017, instead we use the single best hypotheses produced by the best performing system

in Table 3. In a similar task, (Crego and Senellart, 2016) report translation accuracy gains by employing a neural system trained over a synthetic corpus built from source reference sentences and target translation hypotheses. The authors claim that text simplification is achieved when translating with an automatic engine compared to reference (human) translations, leading to higher accuracy results.

Table 4 details the hyper-specialisation training work. Note that the entire hyper-specialisation process was performed on approximately 6 minutes. We used a learning rate set to 0.7. Further experiments need to be conducted for a better understanding of the learning rate role in hyper-specialisation work.

Epoch	1	1
German→English		
Data	T	T-2017
Time (seconds)	365	305
LR	0.7^1	0.7^1
BLEU (2017)	32, 87	32, 66
English→German		
Data	T	T-2017
Time (seconds)	372	308
LR	0.7^1	0.7^1
BLEU (2017)	26, 98	26, 80

Table 4: *Hyper-specialisation on news test sets.*

Accuracy gains are obtained despite using automatic (noisy) translation hypotheses to hyper-specialise: +0.52 (German→English) and +0.57 (English→German). In order to measure the impact of using newstest2017 as training data (sefl-training) we repeated the hyper-specialisation experiment using as training data newstest sets from 2009 to 2016. This is, excluding newstest2017 (**T-2017**). Slightly lower accuracy results were obtained by this second configuration (last column in Table 4) but still outperforming the systems without hyper-specialisation: +0.31 (German→English) and +0.39 (English→German).

4 Conclusions

We described SYSTRAN's submissions to the WMT 2017 shared news translation task for English-German. Our systems are built using OpenNMT. We experimented using monolingual data automatically back-translated. Our resulting models were successfully hyper-specialised with an adaptation technique that finely tunes models according to the evaluation test sentences. Note that all our submitted systems are single networks. No ensemble experiments were carried out, what typically results in higher accuracy results.

Acknowledgements

We would like to thank the anonymous reviewers for their careful reading of the paper and their many insightful comments and suggestions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. Demoed at NIPS 2014: http://lisa.iro.umontreal.ca/mt-demo/. http://arxiv.org/abs/1409.0473.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systran's pure neural machine translation systems. *CoRR* abs/1610.05540. http://arxiv.org/abs/1610.05540.

Josep Maria Crego and Jean Senellart. 2016. Neural machine translation from simplified translations. *CoRR* abs/1612.06139. http://arxiv.org/abs/1612.06139.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Accepted to ACL 2017 Conference Demo Papers*. Association for Computational Linguistics, Vancouver, Canada.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1412–1421. http://aclweb.org/anthology/D15-1166.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*. Association for Computational Linguistics, Uppsala, Sweden, pages 220–224. http://www.aclweb.org/anthology/P10-2041.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. *Proceedings*

of the 54th Annual Meeting of the Association for Computational Linguistics pages 86–96. http://www.aclweb.org/anthology/P16-1009.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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)*. Association for Computational Linguistics, Berlin, Germany, pages 1715–1725. http://www.aclweb.org/anthology/P16-1162.

Christophe Servan, Josep Maria Crego, and Jean Senellart. 2016. Domain specialization: a post-training domain adaptation for neural machine translation. *CoRR* abs/1612.06141. http://arxiv.org/abs/1612.06141.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ÅĄukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. Technical report, Google. https://arxiv.org/abs/1609.08144.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *CoRR* abs/1409.2329. http://arxiv.org/abs/1409.2329.