[EAMT May 2010 St Raphael, France]

Lattice Score Based Data Cleaning For Phrase-Based Statistical Machine Translation

Jie Jiang[†], **Andy Way**[†], **Julie Carson-Berndsen**[‡]
[†]CNGL, School of Computing, Dublin City University, Glasnevin, Dublin 9
[‡]Department of Computer Science, University College Dublin, Belfield, Dublin 4
{jjiang,away}@computing.dcu.ie, julie.berndsen@ucd.ie

Abstract

Statistical machine translation relies heavily on parallel corpora to train its models for translation tasks. While more and more bilingual corpora are readily available, the quality of the sentence pairs should be taken into consideration. This paper presents a novel lattice score-based data cleaning method to select proper sentence pairs from the ones extracted from a bilingual corpus by the sentence alignment methods. The proposed method is carried out as follows: firstly, an initial phrase-based model is trained on the full sentence-aligned corpus; then for each of the sentence pairs in the corpus, word alignments are used to create anchor pairs and source-side lattices; thirdly, based on the translation model, target-side phrase networks are expanded on the lattices and Viterbi searching is used to find approximated decoding results; finally, BLEU score thresholds are used to filter out the low-score sentence pairs for the data cleaning purpose. Our experiments on the FBIS corpus showed improvements of BLEU score from 23.78 to 24.02 in Chinese-English.

1 Introduction

To overcome problems of data sparseness, most statistical machine translation (SMT) methods tend to use the largest possible corpora to train the models. Following the word-based (Brown et al., 1993) and phrase-based (Koehn et al., 2003) methods, incorporating as much parallel corpora as possible normally obtains a better system performance. Even in some SMT evaluation tasks (Zollmann et al., 2008), a huge amount (nearly 10M) of sentence pairs are provided to train the models.

However, as the size of the corpus increases dramatically, sentence alignment can only be performed automatically. Since it is inevitable that (a possibly large number of) misaligned sentences will be introduced during the automatic sentence alignment phase, some part of the bilingual corpus will not contribute at all to the SMT systems. Another drawback of this paradigm is that it takes a long time to train models on the full corpus. Furthermore, even if the models are ready for use, sometimes they will become too big to fit into memory, and in addition the decoding speed will suffer.

To overcome these problems, two possible methods may be useful for improving conventional SMT methods: (i) phrase table pruning techniques (Johnson et al., 2007; Yang et al., 2009), which are carried out on the model side; and (ii) data cleaning methods, which are carried out *ab initio* on the training data side.

Some work has been reported on data cleaning of SMT. In (Lü et al., 2007), information retrieval methods are utilized to weight the training data to obtain significant improvements over a baseline SMT system. In (Okita, 2009), both word-based and phrase-based models are trained to decode all the training data, and then the sentence pairs are filtered by various evaluation scores of the decoding results. On the other hand, a lot of work has been done on the learning capabilities of SMT system: by examining system performance under different conditions, (Turchi et al., 2008) argues that SMT systems may not be improved by adding more data in i.i.d ways. Active learning (Haffari et al., 2009) are also introduced to obtain improved performance compared with a random sentence selection scheme.

Another work related to this paper introduces a constraint satisfaction approach (Canisiu et al., 2009), which integrates many different solutions to aspects of the output space. It uses an optimized objective function during the decoding of a word-based SMT system to obtain better translation results.

The starting point of this paper is fairly close to that of (Okita, 2009). However, in there, the pure decoding method suffers from the problem that the decoding of the whole corpus takes a fairly long time to accomplish, which probably makes it impossible for large corpora. Note that in the speech recognition realm, forced-alignment (Malfrere et al., 2003) instead of decoding, can be used to assess goodness of utterances (Witt, 1999) by given the trained acoustic model and transcriptions. Although we cannot port forced-alignment directly into SMT, the idea presented in this paper is to use trained models to access the goodness of sentence pairs and is motivated by the forced-alignment approach. Instead of directly decoding, we are going to use the ready-for-use word alignments to reduce the search space in the decoding phase, and thus to approximate the final decoding results, which are used to obtain the evaluation scores for data cleaning.

The rest of this paper is organized as follows: In section 2 we give a brief overview of the decoding-based data cleaning method. In section 3 we discuss the use of word alignments to reduce the search space in decoding. Section 4 describes our lattice score-based decoding for data cleaning. Section 5 gives the details of lattice building, and section 6 presents Viterbi decoding on lattices. Experiments and results are carried out in section 7. The conclusion and future work are discussed in section 8.

2 Decoding-based data cleaning

The decoding-based data cleaning method for phrase-based SMT in (Okita, 2009) can be formalized as follows:

First train a phrase-based SMT model M using the training corpus. Then for each sentence pair in the training corpus, where s and t_{ref} are the source and target sentences respectively, decode s by the model M to derive its translation t^*. Then take t_{ref} as the reference to evaluate translation t^*, and use the evaluation scores as the criterion for data cleaning. After the removal of sentence pairs with low evaluation scores (or X-gram scores in (Okita, 2009)), a new model is trained on the cleaned corpus.

The decoding phase works as follows: denote t as the target sentence for s, and let $\sigma_{s,t}$ be the segmentation of s and t. Following the log-linear model (Och & Ney, 2002), the decoding result t^* for s is represented as in (1):

$$t^* = \arg\max_{t,\sigma} \prod_{f \in F} H_f(s,t,\sigma)^{\lambda_f} \qquad (1)$$

where the set F is a finite set of features and λ_f are the weights of the feature functions H_f of the aligned source and target sentence pairs. The set of features F consists of a phrase translation model (phrase translation probability, lexical weighting, phrase penalty), a language model, a distance-based reordering model, the word penalty and a lexicalized reordering model (Koehn et al., 2005).

For the cleaning stage, various evaluation methods, such as BLEU (Papineni et al., 2002), METEOR (Banerjee et al., 2005), NIST (Doddington, 2002) are taken into account to score the decoding result t^* and target-side correspondence in the training corpus t_{ref} of the source sentence s. Finally, those sentence pairs with low scores are filtered out to obtain a cleaned corpus to train the final SMT models.

Since during the decoding process all the possible segmentation and alignments between the source and target sentences are to be examined, it will take quite a long time to perform this method, especially when the training corpus is large. This paper focuses on how to reduce the search space of the decoding phase mentioned above, by taking into account the word alignment information that is readily available after the training phase.

3 The use of word alignments

During the training phase, most phrase-based SMT systems start with the word alignment information, most often using GIZA++,[1] an implementation of the IBM Models. Following the heuristic approach proposed by (Och & Ney, 2003), the bidirectional alignments of the sentence pairs are refined by extending the intersection of the alignment points of the two word alignments by the union of the two

[1] http://fjoch.com/GIZA++.html

word alignments. Finally, the refined word alignments are used to extract phrase pairs (Zens et al., 2002).

In this paper, the word alignments are not only used for extracting phrase pairs, but also to reduce the search space in the decoding process for the data cleaning phase. In formula (1), all possible target sentences and all segmentations of both the source and target sentences are considered during the decoding process. However, since word alignment information has already been extracted for each of the sentence pairs in the training corpus, the search space can be approximated by only allowing the possible target sentences and segmentations that can be inferred from the word alignment information. Given the word alignment A between sentence pair s and t_{ref}, and the translation model M derived from the full corpus, formula (1) can be modified as in (2):

$$\hat{t} = \mathop{\arg\max}_{t \in \hat{T}(M,A), \sigma \in \hat{\Sigma}(A)} \prod_{f \in F} H_f(s,t,\sigma)^{\lambda_f} \tag{2}$$

where $\hat{T}(M,A)$ is the set of all possible target sentences that can be inferred from the translation model M and the word alignment A and $\hat{\Sigma}(A)$ is the set of all possible segmentations that can be inferred from the word alignment A. By using formula (2) to obtain $\hat{t}$ as an approximation of the decoding result t, the data cleaning process can be accomplished by evaluating $\hat{t}$ with t_{ref}. Note that since the search space is dramatically reduced using formula (2), the whole data cleaning process is faster to making it possible for use on large corpora.

4 Lattice score procedure

In the last section, by using the word alignment information extracted in the training phase, the search space is reduced to speed up the decoding of data cleaning. However, it is not necessary to accomplish the decoding procedure in formula (2) to obtain the approximated translation $\hat{t}$ for the source sentence s. Instead of modifying the pruning strategy in the decoding phase, the method proposed in this paper is to use the word alignments to build source-side lattices, and to search on the expanded target phrase networks to extract the hypothesis $\hat{t}$ for a source sentence s. By looking up the translation model, all the features used in the decoding process of formula (1) are utilized to guide the searching. The proposed method involves the following steps:

(1) Collect a set of anchor pairs $\Lambda(A)$ by looking up the word alignment information A for each sentence pairs s and t_{ref} as in (3):

$$\Lambda(A) = \{(\hat{f},\hat{e}) = (f_j^{j+m}, e_i^{i+n}) \mid \forall (i',j') \in A : \\ j \le j' \le j+m \leftrightarrow i \le i' \le i+n\} \tag{3}$$

This formula represents the same procedure with phrase pairs extraction in (Zens et al., 2002), but we use the extracted phrase pair positions $(j,(j+m)-i,(i+n))$ as anchor pairs.

(2) Build a source-side lattice by looking up the target sentence and the target-side positions of the anchor pairs extracted. Nodes and implicit edges of the lattice are created from target words, and non-implicit edges are created from the source words in the anchor pairs $\Lambda(A)$.

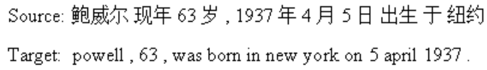
(3) Expand the source-side lattice into target-side phrase network. Nodes and implicit edges are copied into the network, while non-implicit edges are created from the entries of the source words in the translation table.

(4) Search on the target-side phrase network to find a best path with lowest cost of the log-linear model in formula (2). The approximated decoding result $\hat{t}$ can be determined by backtracking the best path and joining the target phrases on the network.

Actually, steps (3) and (4) are carried out together by applying Viterbi algorithm (Young et al., 1989) on the source-side lattice. The details of lattice building and Viterbi decoding will be illustrated in the following two sections.

5 Lattice building

Lattice building process in steps (1) and (2) in the last section is formalized in algorithm 1.


Source: 鲍威尔 现年 63 岁 , 1937 年 4 月 5 日 出生 于 纽约 。

Target: powell , 63 , was born in new york on 5 april 1937 .

Figure 1: Sample sentence pair

Take the sentence pair in Figure 1 as an example. The beginning part of the generated lattice from algorithm 1 is depicted in Figure 2. Note that for illustration purpose, non-implicit edges (with solid lines) in the lattice are labeled by source words f_j^{j+m}, target words e_i^{i+n} (just for illustration) and position information $(j,(j+m)-i,(i+n))$, which

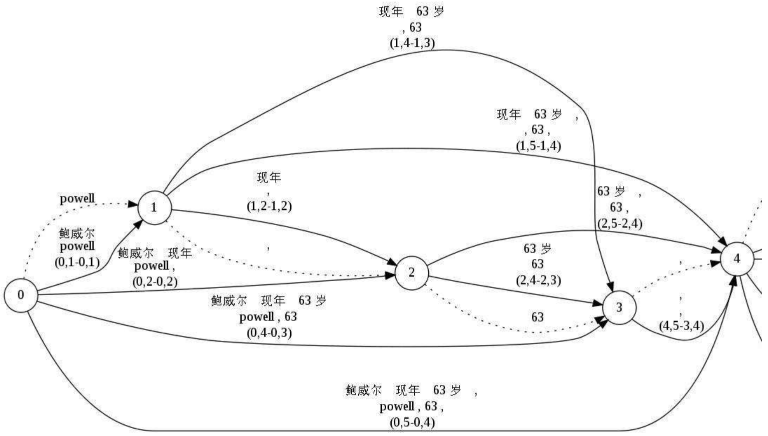


Figure 2: Source-side lattice

```
Algorithm 1 LATTICE BUILDING
Require: Sentence pair and the word alignments.
1: Create anchor pairs $\Lambda$ according to formula (3).
2: Create $N + 1$ nodes according to target sentence length $N$.
3: Connect nodes by $N$ implicit edges labeled by target words.
4: for each anchor pair $(f_j^{j+m}, e_i^{i+n})$ in $\Lambda$ do
5:     Create non-implicit edge $E$ from node $i$ to node $i + n$.
6:     Label $E$ by the source words $f_j^{j+m}$ and the source and target positions $(j, (j + m) - i, (i + n))$
7: end for
```

are displayed in separated lines besides the non-implicit edges. The lattice is constructed as follows: firstly, anchor pairs are extracted from word alignments in the training phase, as is described in line 1 of algorithm 1. Then Nodes are created according to target sentence length. Thirdly, implicit edges (with dotted lines) in figure 2 are created from target words. For example, the implicit edge from node 0 to 1 denote the first word *powell* in the target sentence. After that, non-implicit edges are created from anchor pairs and sentence pairs. For example, for anchor pair position $(0, 4 - 0, 3)$, an non-implicit edge is created from node 0 to 3, labeled with source words from position 0 to 4 (in Figure 2).

6 Viterbi algorithm

The source-side lattices built in the last section are guaranteed to be directed acyclic graphs and their nodes are already in topological order. We apply the token passing implementation of Viterbi algorithm (Young et al., 1989) to carry out the target-side phrase network expanding and the Viterbi searching on the network. Each node in the source-side lattice has a stack to hold tokens. Tokens hold the cost of each partial path together with the back-track information, and can move from one node to another. The searching process is formalized in algorithm 2.

By joining the hypothesis collected from the best path (line 16 of algorithm 2), we can obtain the approximated translation result $\hat{t}$ in formula (2).

Note that some of the nodes do not have any outgoing non-implicit edges, (for example, nodes 3 and 5 in Figure 3), during the searching phase, all tokens on these nodes cannot be propagated any further because there is no non-implicit edge available. Depending on the lattice topology, sometimes there is no path consisting only of non-implicit edges. In this case there will be no Viterbi search result, and no target translation. To overcome this, implicit edges are utilized to accomplish the searching. Token generation and costs via implicit edges are discussed in section 6.1.

6.1 Token costs

There are two kinds of token costs in algorithm 2:

(1) Normal cost (line 8 in algorithm 2): It is similar to the log-linear model in formula (2), including scores from the phrase translation models, the language model, word penalties, distance-based and lexical reordering models. Note that since source positions is preserved from the source-side lattice (for example, in Figure 2, first two num-

Algorithm 2 TOKEN PASSING DECODING

```
Require: Source-side lattice, trained phrase-based models.
 1: Add a initial token with zero cost to the start node.
 2: for each node $A$ in the source-side lattice do
 3:   for each token in the stack of $A$ do
 4:     if $A$ has outgoing non-implicit edges then
 5:       for each outgoing non-implicit edge $E$ (with source words $f$) of $A$ do
 6:         Collect set of phrase pairs $P = \{(f', e')\}$ from translation table, where $f = f'$.
 7:         for each phrase pair $(\bar{f}, \bar{e}) \in P$ do
 8:           Pass a new token to the end node $B$ of $E$, with accumulated cost and hypothesis $\bar{e}$.
 9:         end for
10:       end for
11:     else
12:       Pass a new token to the next node $C$ of $A$, with accumulated cost and hypothesis $e$ (the target word on the implicit edge from $A$ to $C$).
13:     end if
14:   end for
15: end for
16: Traceback from the lowest cost token of the last node.
```

bers of the parenthesized labels of non-implicit edges), reordering scores are calculated by looking up source positions of two consecutive tokens.

(2) Implicit-edge cost (line 12 in algorithm 2): Suppose token τ^{n+1} is passed from token τ^n via an implicit edge. The new cost is estimated from the history of τ^n as in formula (4):

$$cost(\tau^{n+1}) = cost(\tau^n) + \frac{cost(\tau^n) - lmcost(\tau^n)}{n} + lmcost(\tau^{n+1}) + word\_penlaty \quad (4)$$

where $lmcost$ indicates the target language model cost of the hypothesis corresponding with the token, and $word\_penalty$ is the same with that in normal cost. The basic assumption held in formula (4) is that tokens with better history will generate better tokens.

Formula (4) penalizes tokens via implicit edges. For example, in Figure 3, from node 1 to 8, it is obvious that tokens via implicit edges (nodes 3 to 4, 5 to 6, 6 to 7 and 7 to 8) will generate more costs because they have to go through more nodes than tokens via the non-implicit edge from node 1 to 8. Note that cost estimation is biased at the beginning of lattices, because history of the path is too short. However, this problem is not obvious since most of the nodes have outgoing non-implicit edges.

6.2 Merging and Pruning

Keeping all the tokens generated in the searching process wastes a significant amount of computing resources. To overcome this, path merging and pruning are adopted to reduce the whole search space. Risk-free merging operation is carried out for tokens on the same node. Only those tokens with same partial translation, same source phrase and same source position are merged to keep the one with lowest cost. Beam pruning is also carried out for all tokens on the same node. While each node in the network has a stack to hold tokens passed from previous nodes, all tokens are sorted in order of path cost. Only a portion of the lower cost tokens are kept using either a probability threshold of the best token or fixed admissible token numbers.

7 Experiments and evaluation

7.1 Experimental settings

The experiments are conducted on Chinese-to-English. Our experimental data come from FBIS corpus, which is a multilingual paragraph aligned corpus with LDC resource number LDC2003E14. Sentence alignment is carried out by Champollion aligner (Ma, 2006), which is designed for noisy data. In this case, our training data are considered as a semi-cleaned corpus, which is a hard task for data cleaning. After the sentence alignment, we have 256,911 sentence pairs as the whole data set. Then 2,000 pairs of development set and 2,000 pairs of test set are selected randomly from the whole data set. After sentence length filtering, the rest of the data set is used as the training set.

We use Moses (Koehn et al., 2007) as the baseline system. The GIZA++ toolkit is used to perform word alignment and "grow-diag-final" refinement method is adopted (Koehn et al., 2003). Phrase extraction is carried out by the method of (Zens et al., 2002), which is also used for lattice generation in this paper. Minimum error rate training (Och, 2003) is performed for tuning. The language model in all experiments is a 5-gram language model trained on the training set

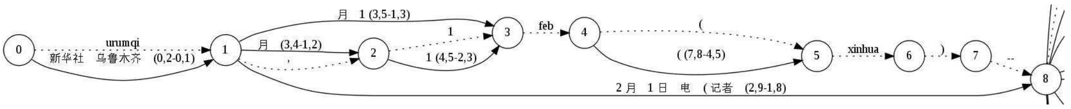


Figure 3: Tokens via implicit edges

using SRILM [2] toolkit with modified Kneser-Ney smoothing (Kneser & Ney, 1995).

Our proposed data cleaning method is carried out using algorithm 3:

Algorithm 3 DATA CLEANING

```
1: Train and tune the baseline Moses model $M$, and obtain the refined word alignments $A$ of the training set.
2: for each sentence pair $(s, t_{ref})$ in the training set do
3:     Use our proposed method in Section 4 to calculate the approximated decoding result $\hat{t}$.
4:     Calculate sentence BLEU score of $\hat{t}$ taking $t_{ref}$ as the reference.
5: end for
6: Filter out all sentence pairs with BLEU scores lower than our predefined threshold from the training.
7: Use filtered training set to re-train and re-tune the Moses model for evaluation.
```

It is worth noting that the baseline translation model, reordering model, language model, tuned weights and word alignments are used in our proposed data cleaning method. No extra training or tuning is needed. In this paper, the filtering thresholds are selected from the distribution of BLEU scores calculated by our lattice score method.

7.2 Results

On the training set, the BLEU scores calculated from the approximated decoding results and the target sentences are depicted in Figure 4. The x-axis indicates the BLEU score and y-axis indicates number of sentences. The mean BLEU score of all lattice score results on the training set is 48.49. From the figure, we note that only a small amount of sentence pairs have a lower BLEU score. Then we take gradually changed BLEU thresholds to filter out sentence pairs with lower scores to check the effectiveness of our lattice score based data cleaning method.

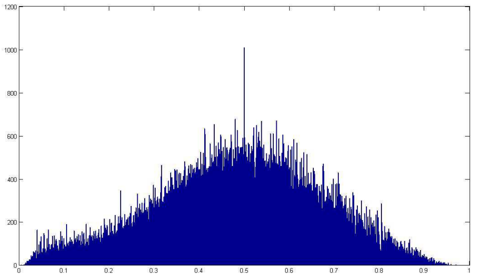


Figure 4: BLEU distribution of lattice score results

By adjusting Figure 4, we adopted four thresholds of BLEU scores for the data cleaning experiments: 10.00, 15.00, 20.00 and 25.00, which removed 2.6%, 4.8%, 7.5% and 11.3% sentences from the original training corpus respectively.

The baseline system and data cleaning systems with four thresholds are illustrated in Tables 1 and 2, for dev set and test set respectively. The phrase table limit is set to 20 and histogram beam is set to 50 in our Viterbi searching on lattices.

Methods	**BLEU**	**NIST**	**METEOR**
baseline	21.73	6.33	51.6
thd=10.00	**21.99**	**6.35**	51.58
thd=15.00	**21.81**	6.33	51.41
thd=20.00	**21.82**	**6.35**	51.47
thd=25.00	21.70	6.25	50.9

Table 1: Results on dev set

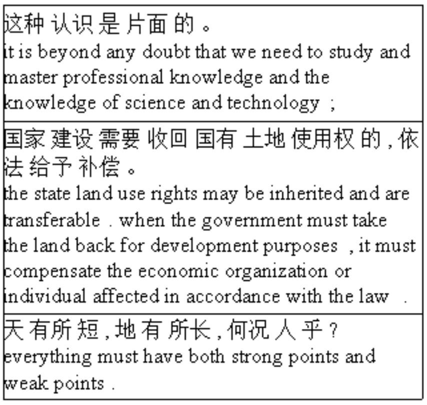
Methods	**BLEU**	**NIST**	**METEOR**
baseline	23.78	6.58	54.07
thd=10.00	**24.02**	**6.60**	**54.31**
thd=15.00	23.77	6.56	**54.14**
thd=20.00	**23.95**	**6.60**	**54.11**
thd=25.00	**23.86**	6.52	53.66

Table 2: Results on test set

As we can see from Tables 1 and 2, our proposed data cleaning method gives positive results over the baseline system with various thresholds. The maximum improvement on test set is 0.24 BLEU

[2] http://www.speech.sri.com/projects/srilm/

score (1% relative improvement), and it is obtained by setting the threshold at 10.00.



这种 认识 是 片面 的 。 it is beyond any doubt that we need to study and master professional knowledge and the knowledge of science and technology ;
国家 建设 需要 收回 国有 土地 使用权 的 , 依法 给予 补偿 。 the state land use rights may be inherited and are transferable . when the government must take the land back for development purposes , it must compensate the economic organization or individual affected in accordance with the law .
天 有所 短 , 地 有 所长 , 何况 人 乎 ? everything must have both strong points and weak points .

Figure 5: Sample sentence pairs

By manually analyze part of the sentence pairs filtered out in the line 6 of algorithm 3, we find that they can be categorized into four kinds of sentences:

- Misaligned pairs: sentence pairs with completely difference meanings. For example, the sentence pair in the first row of Figure 5.
- Partially aligned pairs: one sentence should be aligned to one part of another sentence. For example, in the second row of Figure 5, the source sentence should be aligned to the latter half of the target sentence.
- Garbage pairs: sentence pairs consists of garbage words, which is typically generated from incorrect encodings.
- Hard pairs: sentence pairs that are hard to translate. For example, in the sentence pair in the third row of Figure 5, the source sentence is ancient Chinese which is different from most of the source sentences (modern Chinese) in the corpus.

It is obvious that the first three kinds of sentence pairs are supposed to be removed from the training corpus, because they would not contribute to the final SMT system due to the bad word alignments and phrase extraction on those sentence pairs. To some extent, hard pairs should be cleaned from the training set as well. In our case, only a small portion of sentences in the corpus are ancient Chinese, and they can be treated as another language which is different from the source language, thus, those sentences generate incorrect word alignments and phrase extractions, and SMT system would not benefit from them.

8 Conclusion and future work

In this paper, we introduced a novel lattice score-based data cleaning method to select proper sentence pairs from the sentence-aligned corpus. The procedure is based on conventional phrase-based SMT training and tuning, but the word alignment information is utilized to reduce the search space during the decoding of the training set for data cleaning. Source-side lattice is created according to the anchor pairs extracted from word alignments. Target-side phrase network is expanded from source-side lattice by looking up the phrase table. Token passing is used to search for the best path in the target-side phrase network to obtain the approximated decoding results. Experiments on FBIS data showed the benefits of our data cleaning methods with various BLEU score-based thresholds against baseline Moses system.

In the future, firstly we plan to apply our algorithm on large-scale noisy data and other language pairs. And we will find a better way to adjust the biased cost estimation for tokens via implicit edges at the beginning of lattices.

9 Acknowledgements

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University. Thanks to Yanjun Ma for the sentence-aligned FBIS corpus.

References

Satanjeev Banerjee, Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation With Improved Correlation With Human Judgments. *ACL-2005: Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65-72, Ann Arbor, MI.

Peter F. Brown, Vincent J.D. Pietra, Stephen A.D. Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation *Computational Linguistics*, 19(2):263-311.

Sander Canisius, and Antal van den Bosch 2009. A Constraint Satisfaction Approach to Machine Translation. *EAMT-2009: Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, pp.182-189, Barcelona, Spain.

George Doddington. 2005. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. *HLT 2002: Human Language Technology Conference: proceedings of the second international conference on human language technology research*, pp. 138-145, San Diego, CA.

Gholamreza Haffari, Maxim Roy, and Anoop Sarkar 2009. Active learning for statistical phrase-based machine translation. *NAACL HLT 2009. Human Language Technologies: the 2009 annual conference of the North American Chapter of the ACL*, pp. 415-423, Boulder, Colorado.

Howard Johnson, Joel Martin, George Foster and Roland Kuhn. 2007. Improving Translation Quality by Discarding Most of the Phrasetable. *EMNLP-CoNLL-2007: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 967-975, Prague, Czech Republic.

Reinhard Kneser, Hermann Ney. 1995. Improved Backing-Off for M-Gram Language Modeling. *In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 181-184.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. *In Proceedings of the Human Language Techonology Conference and the North American Association for Computational Linguistics*, pages 127-133, Edmonton, Canada.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. *International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation [IWSLT 2005]*, Pittsburgh, PA, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *ACL 2007: proceedings of demo and poster sessions*, pp. 177-180, Prague, Czech Republic.

Yajuan Lü, Jin Huang and Qun Liu. 2007. Improving Statistical Machine Translation Performance by Training Data Selection and Optimization. *EMNLP-CoNLL-2007: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 343-350, Prague, Czech Republic.

Xiaoyi Ma. 2006. Champollion: A Robust Parallel Text Sentence Aligner. *LREC 2006: Fifth International Conference on Language Resources and Evaluation*, pp.489-492, Genova, Italy.

F. Malfrere, O. Deroo, T. Dutoit, and C. Ris. 2003. Phonetic Alignment: Speech Synthesis-Based vs. Viterbi-Based. *Speech Communication*, 40(4):503-515.

Franz Josef Och, Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, pp. 295-302, Philadelphia, PA.

Franz Josef Och, Hermann Ney. 2007. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19-52.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. *ACL-2003: 41st Annual meeting of the Association for Computational Linguistics*, pp. 160-167, Sapporo, Japan.

Tsuyoshi Okita. 2009. Data Cleaning for Word Alignment. *Proceedings of the ACL-IJCNLP Student Research Workshop*, pp.72-80, Suntec, Singapore.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method For Automatic Evaluation of Machine Translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, pp.311-318, Philadelphia, PA.

Marco Turchi, Tijl de Bie, and Nello Cristianini. 2006. Learning performance of a machine translation system: a statistical and computational analysis. *ACL-08: HLT. Third Workshop on Statistical Machine Translation, Proceedings*, pp. 35-43, Columbus, Ohio, USA.

Silke Maren Witt. 1999. Use of Speech Recognition in Computer-assisted Language Learning. *PhD Thesis. Department of Engineering, University of Cambridge.*

Mei Yang, Jing Zheng. 2009. Toward Smaller, Faster, and Better Hierarchical Phrase-based SMT. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP, Short Papers*, pp.237-240, Suntec, Singapore.

Steve J. Young, N. H. Russell, and J. H. S. Thornton. 1989. Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems. *Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Department*, Cambridge, UK.

Richard Zens, Franz Josef Och, and Hermann Ney. 2002. Phrase-based statistical machine translation. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP*, pp.333-341, Suntec, Singapore.

Andreas Zollmann, Ashish Venugopal, Franz Och, and Jay Ponte. 2008. A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT. *Coling 2008: 22nd International Conference on Computational Linguistics, Proceedings of the conference*, pp.1145-1152, Manchester UK.