Incorporating Source-Language Paraphrases into Phrase-Based SMT with Confusion Networks

Jie Jiang,[†] **Jinhua Du,**[‡] and **Andy Way**[†]

[†]CNGL, School of Computing, Dublin City University, Glasnevin, Dublin 9, Ireland
{jjiang, away}@computing.dcu.ie

[‡]School of Automation and Information Engineering,
Xi'an University of Technology, Xi'an, Shaanxi, China
jhdu@xaut.edu.cn

Abstract

To increase the model coverage, source-language paraphrases have been utilized to boost SMT system performance. Previous work showed that word lattices constructed from paraphrases are able to reduce out-of-vocabulary words and to express inputs in different ways for better translation quality. However, such a word-lattice-based method suffers from two problems: 1) path duplications in word lattices decrease the capacities for potential paraphrases; 2) lattice decoding in SMT dramatically increases the search space and results in poor time efficiency. Therefore, in this paper, we adopt word confusion networks as the input structure to carry source-language paraphrase information. Similar to previous work, we use word lattices to build word confusion networks for merging of duplicated paths and faster decoding. Experiments are carried out on small-, medium- and large-scale English–Chinese translation tasks, and we show that compared with the word-lattice-based method, the decoding time on three tasks is reduced significantly (up to 79%) while comparable translation quality is obtained on the large-scale task.

1 Introduction

With the rapid development of large-scale parallel corpus, research on data-driven SMT has made good progress to the real world applications. Currently, for a typical automatic translation task, the SMT system searches and exactly matches the input sentences with the phrases or rules in the models. Obviously, if the following two conditions could be satisfied, namely:

- the words in the parallel corpus are highly aligned so that the phrase alignment can be performed well;
- the coverage of the input sentence by the parallel corpus is high;

then the "exact phrase match" translation method could bring a good translation.

However, for some language pairs, it is not easy to obtain a huge amount of parallel data, so it is not that easy to satisfy these two conditions. To alleviate this problem, paraphrase-enriched SMT systems have been proposed to show the effectiveness of incorporating paraphrase information. In terms of the position at which paraphrases are incorporated in the MT-pipeline, previous work can be organized into three different categories:

- Translation model augmentation with paraphrases (Callison-Burch et al., 2006; Marton et al., 2009). Here the focus is on the translation of unknown source words or phrases in the input sentences by enriching the translation table with paraphrases.
- Training corpus augmentation with paraphrases (Bond et al., 2008; Nakov, 2008a; Nakov, 2008b). Paraphrases are incorporated into the MT systems by expanding the training data.
- Word-lattice-based method with paraphrases (Du et al., 2010; Onishi et al.,

2010). Instead of augmenting the translation table, source-language paraphrases are constructed to enrich the inputs to the SMT system. Another directly related work is to use word lattices to deal with multi-source translation (Schroeder et al., 2009), in which paraphrases are actually generated from the alignments of difference source sentences.

Comparing these three methods, the word-lattice-based method has the least overheads because:

- The translation model augmentation method has to re-run the whole MT pipeline once the inputs are changed, while the word-lattice-based method only need to transform the new input sentences into word lattices.

- The training corpus augmentation method requires corpus-scale expansion, which drastically increases the computational complexity on large corpora, while the word-lattice-based method only deals with the development set and test set.

In (Du et al., 2010; Onishi et al., 2010), it is also observed that the word-lattice-based method performed better than the translation model augmentation method on different scales and two different language pairs in several translation tasks. Thus they concluded that the word-lattice-based method is preferable for this task.

However, there are still some drawbacks for the word-lattice-based method:

- In the lattice construction processing, duplicated paths are created and fed into SMT decoders. This decreases the paraphrase capacity in the word lattices. Note that we use the phrase "paraphrase capacity" to represent the amount of paraphrases that are actually built into the word lattices. As presented in (Du et al., 2010), only a limited number of paraphrases are allowed to be used while others are pruned during the construction process, so duplicate paths actually decrease the number of paraphrases that contribute to the translation quality.

- The lattice decoding in SMT decoder have a very high computational complexity which makes the system less feasible in real time application.

Therefore, in this paper, we use confusion networks (CNs) instead of word lattices to carry paraphrase information in the inputs for SMT decoders. CNs are constructed from the aforementioned word lattices, while duplicate paths are merged to increase paraphrase capacity (e.g. by admitting more non-duplicate paraphrases without increasing the input size). Furthermore, much less computational complexity is required to perform CN decoding instead of lattice decoding in the SMT decoder. We carried out experiments on small-, medium- and large-scale English–Chinese translation tasks to compare against a baseline PBSMT system, the translation model augmentation of (Callison-Burch et al., 2006) method and the word-lattice-based method of (Du et al., 2010) to show the effectiveness of our novel approach.

The motivation of this work is to use CN as the compromise between speed and quality, which comes from previous studies in speech recognition and speech translation: in (Hakkani-Tür et al., 2005), word lattices are transformed into CNs to obtain compact representations of multiple aligned ASR hypotheses in speech understanding; in (Bertoldi et al., 2008), CNs are also adopted instead of word lattices as the source-side inputs for speech translation systems. The main contribution of this paper is to show that this compromise also works for SMT systems incorporating source-language paraphrases in the inputs.

Regarding the use of paraphrases SMT system, there are still other two categories of work that are related to this paper:

- Using paraphrases to improve system optimization (Madnani et al., 2007). With an English–English MT system, this work utilises paraphrases to reduce the number of manually translated references that are needed in the parameter tuning process of SMT, while preserved a similar translation quality.

- Using paraphrases to smooth translation models (Kuhn et al., 2010; Max, 2010). Either cluster-based or example-based methods are

proposed to obtain better estimation on phrase translation probabilities with paraphrases.

The rest of this paper is organized as follows: In section 2, we present an overview of the word-lattice-based method and its drawbacks. Section 3 proposes the CN-based method, including the building process and its application on paraphrases in SMT. Section 4 presents the experiments and results of the proposed method as well as discussions. Conclusions and future work are then given in Section 5.

2 Word-lattice-based method

Compared with translation model augmentation with paraphrases (Callison-Burch et al., 2006), word-lattice-based paraphrasing for PBSMT is introduced in (Du et al., 2010). A brief overview of this method is given in this section.

2.1 Lattice construction from paraphrases

The first step of the word-lattice-based method is to generate paraphrases from parallel corpus. The algorithm in (Bannard and Callison-Burch, 2005) is used for this purpose by pivoting through phrases in the source- and the target- languages: for each source phrase, all occurrences of its target phrases are found, and all the corresponding source phrases of these target phrases are considered as the potential paraphrases of the original source phrase (Callison-Burch et al., 2006). A paraphrase probability $p(e_2|e_1)$ is defined to reflect the similarities between two phrases, as in (1):

$$p(e_2|e_1) = \sum_f p(f|e_1)p(e_2|f) \quad (1)$$

where the probability $p(f|e_1)$ is the probability that the original source phrase e_1 translates as a particular phrase f on the target side, and $p(e_2|f)$ is the probability that the candidate paraphrase e_2 translates as the source phrase. Here $p(e_2|f)$ and $p(f|e_1)$ are defined as the translation probabilities estimated using maximum likelihood by counting the observations of alignments between phrases e and f in the parallel corpus, as in (2) and (3):

$$p(e_2|f) \approx \frac{count(e_2, f)}{\sum_{e_2} count(e_2, f)} \quad (2)$$

$$p(f|e_1) \approx \frac{count(f, e_1)}{\sum_{f} count(f, e_1)} \quad (3)$$

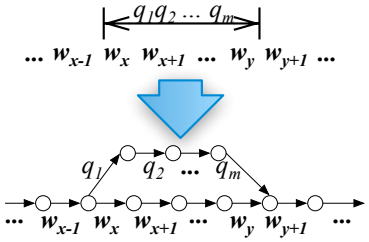

Figure 1: Construct word lattices from paraphrases.

The second step is to transform input sentences in the development and test sets into word lattices with paraphrases extracted in the first step. As illustrated in Figure 1, given a sequence of words $\{w_1, \ldots, w_N\}$ as the input, for each of the paraphrase pairs found in the source sentence (e.g. $p_i = \{q_1, \ldots, q_m\}$ for $\{w_x, \ldots, w_y\}$), add in extra nodes and edges to make sure those phrases coming from paraphrases share the same start nodes and end nodes with that of the original ones. Subsequently the following empirical methods are used to assign weights on paraphrases edges:

- Edges originating from the input sentences are assigned weight 1.
- The first edges for each of the paraphrases are calculated as in (4):

$$w(e^1_{p_i}) = \frac{1}{k+i} \ \ (1 <= i <= k) \quad (4)$$

  where 1 stands for the first edge of paraphrase p_i, and i is the probability rank of p_i among those paraphrases sharing with a same start node, while k is a predefined constant as a trade-off parameter for efficiency and performance, which is related to the paraphrase capacity.
- The rest of the edges corresponding to the paraphrases are assigned weight 1.

The last step is to modify the MT pipeline to tune and evaluate the SMT system with word lattice inputs, as is described in (Du et al., 2010; Onishi et al., 2010).

For further discussion, a real example of the generated word lattice is illustrated in Figure 2. In the word lattice, double-line circled nodes and solid lined edges come from originated from the original sentence, while others are generated from paraphrases. Word, weight and ranking of each edge are displayed in the figure. By adopting such an input structure, the diversity of the input sentences is increased to provide more flexible translation options during the decoding process, which has been shown to improve translation performance (Du et al., 2010).

2.2 Path duplication and decoding efficiency

As can be seen in Figure 2, the construction process in the previous steps tends to generate duplicate paths in the word lattices. For example, there are two paths from node 6 to node 11 with the same words "secretary of state" but different edge probabilities (the path via node 27 and 28 has the probability $1/12$, while the path via node 26 and 9 has the probability $1/99$). This is because the aforementioned straightforward construction process does not track path duplications from different spans on the source side. Since the number of admitted paraphrases is restricted by parameter k in formula (4), the path duplication will decrease the paraphrase capacity to a certain extend.

Moreover, state of the art PBSMT decoders (e.g. Moses (Koehn et al., 2007)) have a much higher computational complexity for lattice structures than for sentences. Thus even though only the test sentences need to be transformed into word lattices, decoding time is still too slow for real-time applications.

Motivated by transforming ASR word-graphs into CNs (Bertoldi et al., 2008), we adopt CN as the trade-off between efficiency and quality. We aim to merge duplicate paths in the word lattices to increase paraphrase capacity, and to speed up the decoding process via CN decoding. Details of the proposed method are presented in the following section.

3 Confusion-network-based method

CNs are weighted direct graphs where each path from the start node to the end node goes through all the other nodes. Each edge is labelled with a word and a probability (or weight). Although it is commonly required to normalize the probability of edges between two consecutive nodes to sum up to one, from the point of view of the decoder, this is not a strict constraint as long as any score is provided (similar to the weights on the word lattices in the last section, and we prefer to call it "weight" in this case).

The benefits of using CNs are:

1. the ability to represent the original word lattice with a highly compact structure;

2. all hypotheses in the word lattice are totally ordered, so that the decoding algorithm is mostly retained except for the collection of translation options and the handeling of ϵ edges (Bertoldi et al., 2008), which requires much less computational resources than the lattice decoding.

The rest of this section details the construction process of the CNs and the application in paraphrase-enriched SMT.

3.1 Confusion Network building

We build our CN from the aforementioned word lattices. Previous studies provide several methods to do this. (Mangu et al., 2000) propose a method to cluster lattice words on the similarity of pronunciations and frequency of occurrence, and then to create CNs using cluster orders. Although this method has a computational complexity of $O(n^3)$, the SRILM toolkit (Stolcke, 2002) provides a modified algorithm which runs much faster than the original version. In (Hakkani-Tür et al., 2005), a pivot algorithm is proposed to form CNs by normalizing the topology of the input lattices.

In this paper, we use the modified method of (Mangu et al., 2000) provided by the SRILM toolkit to convert word lattices into CNs. Moreover, we aim to obtain CNs with the following guidelines:

- Cluster the lattice words only by topological orders and edge weights without considering word similarity. The objective is to reduce the

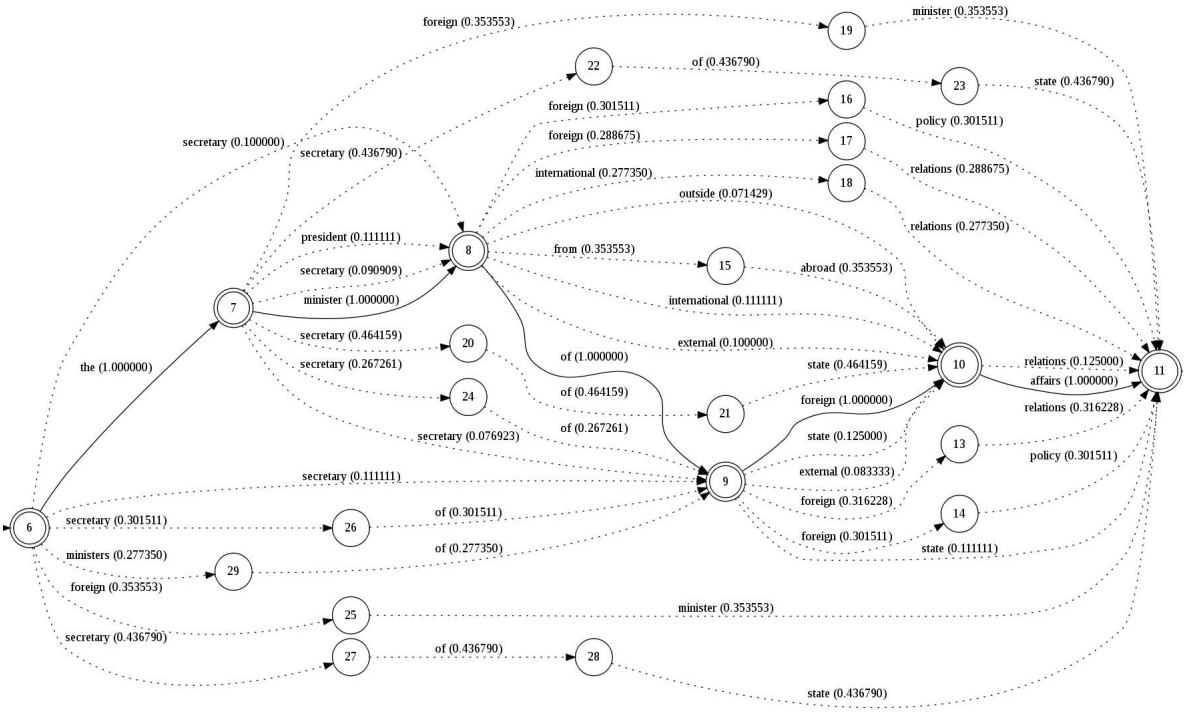

Figure 2: An example of a real paraphrase lattice. Note that it is a subsection of the whole word lattice that is too big to fit into this page, and edge weights have been evenly distributed for CN conversion as specified by formula (5).

impact of path duplications in the building process, since duplicate words will bias the importance of paths.

- Assign edge weights by the ranking of paraphrase probabilities, rather than by posterior probabilities from the modified method of (Mangu et al., 2000). This is similar to that given in formula (4). The reason for this is to reduce the impact of path duplications on the calculation of weights.

Thus, we modified the construction process as follows:

1. For each of the input word lattices, replace word texts with unique identifiers (to make the lattice alignment uncorrelated to the word similarity, since in this case, all words in the lattice are different from each other).

2. Evenly distribute edge weights for each of the lattices by modifying formula (4) as in (5):

$$w(e_{p_i}^j) = \frac{1}{\sqrt[M_i]{(k+i)}} \ (1 <= i <= k) \quad (5)$$

where $1 <= j <= M_i$, given $e_{p_i}^j$ is the j^{th} edge of paraphrase p_i, and M_i is the number of words in p_i. This is to avoid large weights on the paraphrase edges for lattice alignments.

3. Transform the weighted word lattices into CNs with the SRILM toolkit, and the paraphrase ranking information is carried on the edges.

4. Replace the word texts in step 1, and then for each column of the CN, merge edges with same words by keeping those with the highest ranking (a smaller number indicates a higher ranking, and edges from the original sentences will always have the highest ranking). Note that to assign ranking for each ϵ edge which does not appear in the word lattice, we use the ranking of non-original edges (in the same column) which have the closest posterior probability to it. (Assign ranking 1 if failed to find a such edge).

5. Reassign the edge weights: 1) edges from original sentences are assigned with weight 1; 2) edges from paraphrases are assigned with an

empirical method as in (6):

$$w(e_{p_i}^{cn}) = \frac{1}{k+i} \ \ (1 <= i <= k) \qquad (6)$$

where $e_{p_i}^{cn}$ are edges corresponding with paraphrase p_i, and i is the probability rank of p_i in formula (4), while k is also defined in formula (4).

A real example of a constructed CN is depicted in Figure 3, which is correspondent with the word lattice in Figure 2. Unlike the word lattices, all the nodes in the CN are generated from the original sentence, while solid lined edges come from the original sentence, and dotted lined edges correspond to paraphrases.

As in shown in the Figures, duplicate paths in the word lattices have been merged into CN edges by step 4. For example, the two occurrences of "secretary of state" in the word lattices (one path from node 6 to 11 via 27 and 28, and one path from node 6 to 11 via 26 and 9 in the word lattice) are merged to keep the highest-ranked path in the CN (note there is one ϵ edge between node 9 and 10 to accomplish the merging operation). Furthermore, each edge in the CN is assigned a weight by formula (6). This weight assignment procedure penalizes paths from paraphrases according to the paraphrase probabilities, in a similar manner to the aforementioned word-lattice-based method.

3.2 Modified MT pipeline

By transforming word lattices into CNs, duplicate paths are merged. Furthermore the new features on the edges are introduced by formula (6), which is then tuned on the development set using MERT (Och, 2003) in the log-linear model (Och and Ney, 2002). Since the SMT decoders are able to perform CN decoding (Bertoldi et al., 2008) in an efficient multi-stack decoding way, decoding time is drastically reduced compared to lattice decoding.

The training steps are then modified as follows: 1) Extract phrase table, reordering table, and build target-side language models from parallel and monolingual corpora respectively for the PBSMT model; 2) Transform source sentences in the development set into word lattices, and then transform them into CNs using the method proposed in Section 3.1; 3) Tune the PBSMT model on the CNs via the development set. Note that the overhead of the evaluation steps are: transform each test set sentence into a word lattice, and also transform them into a CN, then feed them into the SMT decoder to obtain decoding results.

4 Experiments

4.1 Experimental setup

Experiments were carried out on three English–Chinese translation tasks. The training corpora comprise 20K, 200K and 2.1 million sentence pairs, where the former two corpora are derived from FBIS corpus[1] which is sentence-aligned by Champollion aligner (Ma, 2006), the latter corpus comes from HK parallel corpus,[2] ISI parallel corpus,[3] other news data and parallel dictionaries from LDC.

The development set and the test set for the 20K and 200K corpora are randomly selected from the FBIS corpus, each of which contains 1,200 sentences, with one reference. For the 2.1 million corpus, the NIST 2005 Chinese–English current set (1,082 sentences) with one reference is used as the development set, and NIST 2003 English–Chinese current set (1,859 sentences) with four references is used as the test set.

Three baseline systems are built for comparison: Moses PBSMT baseline system (Koehn et al., 2007), a realization of the translation model augmentation system described in (Callison-Burch et al., 2006) (named "Para-Sub" hereafter), and the word-lattice based system proposed in (Du et al., 2010).

Word alignments on the parallel corpus are performed using GIZA++ (Och and Ney, 2003) with the "grow-diag-final" refinement. Maximum phrase length is set to 10 words and the parameters in the log-linear model are tuned by MERT (Och, 2003). All the language models are 5-gram built with the SRILM toolkit (Stolcke, 2002) on the monolingual part of the parallel corpora.

4.2 Paraphrase acquisition

The paraphrases data for all paraphrase-enriched system is derived from the "Paraphrase Phrase Ta-

[1]Paragraph-aligned corpus with LDC number LDC2003E14.

[2]LDC number: LDC2004T08.

[3]LDC number: LDC2007T09.

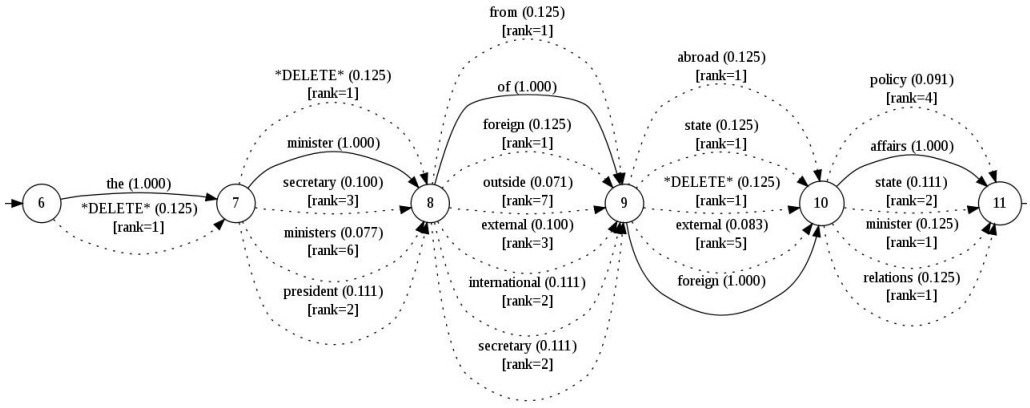

Figure 3: An example of a real CN converted from a paraphrase lattice. Note that it is a subsection of the whole CN that is converted from the word lattice in Figure 2.

ble"[4] of TER-Plus (Snover et al., 2009). Furthermore, the following two steps are taken to filter out noise paraphrases as described in (Du et al., 2010):

1. Filter out paraphrases with probabilities lower than 0.01.

2. Filter out paraphrases which are not observed in the phrase table. This objective is to guarantee that no extra out-of-vocabulary words are introduced into the paraphrase systems.

The filtered paraphrase table is then used to generate word lattices and CNs.

4.3 Experimental results

The results are reported in BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) scores.

Table 1 compares the performance of four systems on three translation tasks. As can be observed from the Table, for 20K and 200K corpora, the word-lattice-based system accomplished the best results. For the 20K corpus, the CN outperformed the baseline PBSMT by 0.31 absolute (2.15% relative) BLEU points and 1.5 absolute (1.99% relative) TER points. For the 200K corpus, it still outperformed the "Para-Sub" by 0.06 absolute (0.26% relative) BLEU points and 0.15 absolute (0.23% relative) TER points. Note that for the 2.1M corpus, although CN underperformed the best word lattice by an insignificant amount (0.06 absolute, 0.41% relative) in terms of BLEU points, it has the best performance in terms of TER points (0.22 absolute, 0.3% relative than word lattice). Furthermore, the CN outperformed "Para-Sub" by 0.36 absolute (2.55% relative) BLEU points and 1.37 absolute (1.84% relative) TER points, and also beat the baseline PBSMT system by 0.45 absolute (3.21% relative) BLEU points and 1.82 absolute (2.43% relative) TER points. The paired 95% confidence interval of significant test (Zhang and Vogel, 2004) between the "Lattice" and "CN" system is [-0.19, +0.38], which also suggests that the two system has a comparable performance in terms of BLEU.

In Table 2, decoding time on test sets is reported to compare the computational efficiency of the baseline PBSMT, word-lattice-based and CN-based methods. Note that word lattice construction time and CN building time (including word lattice construction and conversion from word lattices into CNs with the SRILM toolkit (Stolcke, 2002)) are counted in the decoding time and illustrated in the table within parentheses respectively. Although both word-lattice-based and CN-based methods require longer decoding times than the baseline PBSMT system, it is observed that compared with the word lattices, CNs reduced the decoding time significantly on three tasks, namely 52.06% for the 20K model, 75.75% for the 200K model and 78.88% for the 2.1M model. It is also worth noting that the "Para-Sub" system has a similar decoding time with baseline PBSMT since only the translation table is modified.

[4] http://www.umiacs.umd.edu/~snover/terp/downloads/terp-pt.v1.tgz

System	20K		200K		2.1M	
	BLEU	TER	BLEU	TER	BLEU	TER
Baseline PBSMT	14.42	75.30	23.60	63.65	14.04	74.88
Para-Sub	14.78	73.75	23.41	63.84	14.13	74.43
Word-lattice-based	**15.44**	**73.06**	**25.20**	**62.37**	**14.55**	73.28
CN-based	14.73	73.8	23.47	63.69	14.49	**73.06**

Table 1: Comparison on PBSMT, "Para-Sub", word-lattice and CN-based methods.

System	FBIS testset (1,200 inputs)		NIST testset (1,859 inputs)
	20K model	200K model	2.1M model
Baseline	21 min	41 min	37 min
Lattice	102 min (+ 15 sec)	398 min (+ 20 sec)	559 min (+ 21 sec)
CN	48 min (+ 61 sec)	95 min (+ 96 sec)	116 min (+ 129 sec)

Table 2: Decoding time comparison of PBSMT, word-lattice ("Lattice") and CN-based ("CN") methods.

4.4 Discussion

From the performance and decoding time reported in the last section, it is obvious that on large scale corpora, the CN-based method significantly reduced the computational complexity while preserved the system performance of the best lattice-based method. Thus it makes the paraphrase-enriched SMT system more applicable to real-world applications. On the other hand, for small- and medium-scale data, CNs can be used as a compromise between speed and quality, since decoding time is much less than word lattices, and compared with the "Para-Sub" system, the only overhead is the transforming of the input sentences.

It is also interesting that the relative performance of the CNs increases gradually with the size of the training corpus, which indicates that it is more suitable for models built from large scale data. Considering the decoding time, it is preferable to use CNs instead of word lattices for such translation tasks. However, for the small- and medium-scale data, the CN system is not competitive even compared with the baseline. In this case it suggests that, on these two tasks, the *coverage* issue is not solved by incorporating paraphrases with the CN structure. It might because of the ambiguity that introduced by CNs harms the decoder to choose the appropriate source words from paraphrases. On the other hand, this ambiguity could be decreased with translation models trained on a large corpus, which provides enough observations for the decoders to favour paraphrases.

5 Conclusion and future work

In this paper, CNs are used instead of word lattices to incorporate paraphrases into SMT. Transformation from word lattices into CNs is used to merge path duplications, and decoding time is drastically reduced with CN decoding. Experiments are carried out on small-, medium- and large-scale English–Chinese translation tasks and confirm that compared with word lattices, it is much more computationally efficient to use CNs, while no loss of performance is observed on the large-scale task.

In the future, we plan to apply more features such as source-side language models and phrase length (Onishi et al., 2010) on the CNs to obtain better system performance. Furthermore, we will carry out this work on other language pairs to show the effectiveness of paraphrases in SMT systems. We will also investigate the reason for its lower performance on the small- and medium-scale corpora, as well as the impact of the paraphrase filtering procedure on translation quality.

Acknowledgments

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 the reviewers for their invaluable comments and suggestions.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *43rd Annual meeting of the Association for Computational Linguistics*, Ann Arbor, MI, pages 597–604.

Nicola Bertoldi, Richard Zens, Marcello Federico, and Wade Shen 2008. Efficient Speech Translation Through Confusion Network Decoding. In *IEEE Transactions on Audio, Speech, and Language Processing, 16(8)*, pages 1696–1705.

Francis Bond, Eric Nichols, Darren Scott Appling and Michael Paul. 2008. Improving Statistical Machine Translation by Paraphrasing the Training Data. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Hawaii, pages 150–157.

Chris Callison-Burch, Philipp Koehn and Miles Osborne. 2006. Improved Statistical Machine Translation Using Paraphrases. In *Proceedings of the Human Language Technology conference - North American chapter of the Association for Computational Linguistics (HLT-NAACL)*, NY, pages 17–24.

Jinhua Du, Jie Jiang and Andy Way. 2010. Facilitating Translation Using Source Language Paraphrase Lattices. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Cambridge, MA, pages 420–429.

Dilek Hakkani-Tür, Frédéric Béchet, Giuseppe Riccardi and Gokhan Tur. 2005. Beyond ASR 1-best: Using word confusion networks in spoken language understanding. In *Computer Speech and Language (2005): 20(4)*, pages 495–514.

Philipp Koehn, Hieu Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, Wade Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007: demo and poster sessions*, Prague, Czech Republic, pages 177–180.

Roland Kuhn, Boxing Chen, George Foster and Evan Stratford. 2010. Phrase Clustering for Smoothing TM Probabilities - or, How to Extract Paraphrases from Phrase Tables. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China, pages 608–616.

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

Nitin Madnani, Necip Fazil Ayan, Philip Resnik and Bonnie J. Dorr. 2007. Using Paraphrases for Parameter Tuning in Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, pages 120–127.

Lidia Mangu, Eric Brill and Andreas Stolcke. 2000. Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks. In *Computer Speech and Language 14 (4)*, pages 373–400.

Yuval Marton, Chris Callison-Burch and Philip Resnik. 2009. Improved Statistical Machine Translation Using Monolingually-Derived Paraphrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, pages 381–390.

Aurélien Max. 2010. Example-Based Paraphrasing for Improved Phrase-Based Statistical Machine Translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, pages 656–666.

Preslav Nakov. 2008a. Improved Statistical Machine Translation Using Monolingual Paraphrases In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, Patras, Greece, pages 338–342.

Preslav Nakov. 2008b. Improving English-Spanish statistical machine translation: experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing. In *Proceedings of ACL-08:HLT. Third Workshop on Statistical Machine Translation*, Columbus, Ohio, USA, pages 147–150.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, pages 295–302.

Franz Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics, 29(1)*, pages 19–51.

Takashi Onishi, Masao Utiyama and Eiichiro, Sumita. 2010. Paraphrase Lattice for Statistical Machine Translation. In *Proceedings of the ACL 2010 Conference Short Papers*, Uppsala, Sweden, pages 1–5.

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.

Josh Schroeder, Trevor Cohn and Philipp Koehn. 2009. Word Lattices for Multi-Source Translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, Athens, Greece, pages 719–727.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, pages 223–231.

Matthew Snover, Nitin Madnani, Bonnie J.Dorr and Richard Schwartz. 2009. Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, pages 259–268.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Denver, Colorado, pages 901–904.

Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*. pages 85–94.