

Phrase-Level Combination of SMT and TM Using Constrained Word Lattice

Liangyou Li and Andy Way and Qun Liu

ADAPT Centre, School of Computing

Dublin City University

Dublin 9, Ireland

{liangyouli, away, qliu}@computing.dcu.ie

Abstract

Constrained translation has improved statistical machine translation (SMT) by combining it with translation memory (TM) at sentence-level. In this paper, we propose using a *constrained word lattice*, which encodes input phrases and TM constraints together, to combine SMT and TM at phrase-level. Experiments on English–Chinese and English–French show that our approach is significantly better than previous combination methods, including sentence-level constrained translation and a recent phrase-level combination.

1 Introduction

The combination of statistical machine translation (SMT) and translation memory (TM) has proven to be beneficial in improving translation quality and has drawn attention from many researchers (Biçici and Dymetman, 2008; He et al., 2010; Koehn and Senellart, 2010; Ma et al., 2011; Wang et al., 2013; Li et al., 2014). Among various combination approaches, constrained translation (Koehn and Senellart, 2010; Ma et al., 2011) is a simple one and can be readily adopted.

Given an input sentence, constrained translation retrieves similar TM instances and uses matched segments to constrain the translation space of the input by generating a constrained input. Then an SMT engine is used to search for a complete translation of the constrained input.

Despite its effectiveness in improving SMT, previous constrained translation works at the sentence-level, which means that matched segments in a TM instance are either all adopted or all abandoned regardless of their individual quality (Wang et al., 2013). In this paper, we propose a phrase-level constrained translation approach

which uses a *constrained word lattice* to encode the input and constraints from the TM together and allows a decoder to directly optimize the selection of constraints towards translation quality (Section 2).

We conduct experiments (Section 3) on English–Chinese (EN–ZH) and English–French (EN–FR) TM data. Results show that our method is significantly better than previous combination approaches, including sentence-level constrained methods and a recent phrase-level combination method. Specifically, it improves the BLEU (Papineni et al., 2002) score by up to +5.5% on EN–ZH and +2.4% on EN–FR over a phrase-based baseline (Koehn et al., 2003) and decreases the TER (Snover et al., 2006) error by up to -4.3%/-2.2%, respectively.

2 Constrained Word Lattice

A word lattice $\mathcal{G} = (V, E, \Sigma, \phi, \psi)$ is a directed acyclic graph, where V is a set of nodes, including a start point and an end point, $E \subseteq V \times V$ is a set of edges, Σ is a set of symbols, a label function $\phi : E \rightarrow \Sigma$ and a weight function $\psi : E \rightarrow \mathbb{R}$.¹ A constrained word lattice is a special case of a word lattice, which extends Σ with extra symbols (i.e. constraints).

A constraint is a target phrase which will appear in the final translation. Constraints can be obtained in two ways: *addition* (Ma et al., 2011) and *subtraction* (Koehn and Senellart, 2010).² Figure 1 exemplifies the differences between them.

The construction of a constrained lattice is very similar to that of a word lattice, except that we need to label some edges with constraints. The general process is:

¹In this paper, edge weights are set to 1.

²*Addition* means that constraints are added from a TM target to an input, while *subtraction* means that some constraints are removed from the TM target.

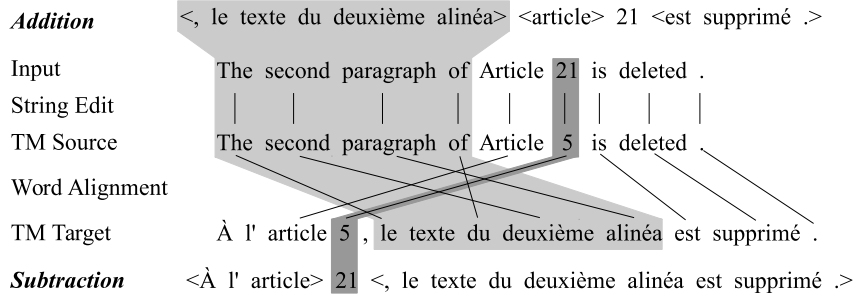


Figure 1: An example of generating a constrained input in two ways: *addition* and *subtraction*. While *addition* replaces an input phrase with a target phrase from a TM instance (an example is marked by lighter gray), *subtraction* removes mismatched target words and inserts mismatched input words (darker gray). Constraints are specified by $\langle \rangle$. Sentences are taken from Koehn and Senellart (2010).

1. Building an initial lattice for an input sentence. This produces a chain.
2. Adding phrasal constraints into the lattice which produces extra nodes and edges.

Figure 2 shows an example of a constrained lattice for the sentence in Figure 1.

In the rest of this section, we explain how to use *addition* and *subtraction* to build a constrained lattice and the decoder for translating the lattice. Notations we use in this section are: an input f and a TM instance $\langle f', e', A \rangle$ where f' is the TM source, e' is the TM target and A is a word alignment between f' and e' .

2.1 Addition

In *addition*, matched input words are directly replaced by their translations from a retrieved TM, which means that *addition* follows the word order of an input sentence. This property makes it easy to obtain constraints for an input phrase.

For an input phrase \bar{f} , we firstly find its matched phrase \bar{f}' from f' via string edits³ between f and f' , so that $\bar{f} = \bar{f}'$. Then, we extract its translation \bar{e}' from e' , which is consistent with the alignment A (Och and Ney, 2004).

To build a lattice using *addition*, we directly add a new edge to the lattice which covers \bar{f} and is labeled by \bar{e}' . For example, dash-dotted lines in Figure 2 are labeled by constraints from *addition*.

³String edits, as used in the Levenshtein distance (Levenshtein, 1966), include match, substitution, deletion, and insertion with a priority in this paper: match > substitution > deletion > insertion.

2.2 Subtraction

In *subtraction*, mismatched input words in f are inserted into e' and mismatched words in e' are removed. The inserted position is determined by A . The advantage of *subtraction* is that it keeps the word order of e' . This is important since the reordering of target words is one of the fundamental problems in SMT, especially for language pairs which have a high degree of syntactic reordering.

However, this property makes it hard to build a lattice from *subtraction*, as – different from the *addition* – *subtraction* does not directly produce a constraint for an input phrase. Thus, for some generated constraints, there is not a specific corresponding phrase in the input. In addition, when adding a constraint to the lattice, we need to consider its context so that the lattice keeps target word order.

To solve this problem, in this paper we propose to segment an input sentence into a sequence of phrases according to information from a matched TM (i.e. the string edit and word alignment) and then create a constrained input for each phrase and add them to the lattice.

Formally, we produce a monotonic segmentation, $\langle \bar{f}_1, \bar{f}'_1, \bar{e}'_1 \rangle \cdots \langle \bar{f}_N, \bar{f}'_N, \bar{e}'_N \rangle$, for each sentence triple: $\langle f, f', e' \rangle$. Each $\langle \bar{f}_i, \bar{f}'_i, \bar{e}'_i \rangle$ tuple is obtained in two phases: (1) According to the alignment A , \bar{f}'_i and \bar{e}'_i are produced. (2) Based on string edits between f and f' , \bar{f}_i is recognized. The resulting tuple is subject to several restrictions:

1. Each $\langle \bar{f}_i, \bar{e}'_i \rangle$ is consistent with the word alignment A and at least one word in \bar{f}_i is aligned to words in \bar{e}'_i .

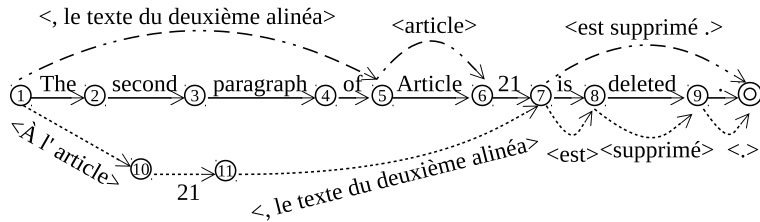


Figure 2: An example of constructing a constrained word lattice for the sentence in Figure 1. Dash-dotted lines are generated by *addition* and dotted lines are generated by *subtraction*. Constraints are specified by $\langle \rangle$.

2. Each boundary word in \bar{f}'_i is either the first word or the last word of f' or aligned to at least one word in e' , so that mismatched input words in \bar{f}'_i which are unaligned can find their position in the current tuple.
3. The string edit for the first word of \bar{f}'_i , where $i \neq 1$, is not “deletion”. That means the first word is not an extra input word. This is because, in *subtraction*, the inserted position of a mismatched unaligned word depends on the alignment of the word before it.
4. No smaller tuples may be extracted without violating restrictions 1–3. This allows us to obtain a unique segmentation where each tuple is minimal.

After obtaining the segmentation, we create a constrained input for each \bar{f}'_i using *subtraction* and add it to the lattice by creating a path covering \bar{f}'_i . The path contains one or more edges, each of which is labeled either by an input word or a constraint in the constrained input.

2.3 Decoding

The decoder for integrating word lattices into the phrase-based model (Koehn et al., 2003) works similarly to the phrase-based decoder, except that it tracks *nodes* instead of words (Dyer et al., 2008): given the topological order of nodes in a lattice, the decoder builds a translation hypothesis from left to right by selecting a range of untranslated nodes.

The decoder for a constrained lattice works similarly except that, for a constrained edge, the decoder can only build its translation directly from the constraint. For example, in Figure 2, the translation of the edge “1 \rightarrow 5” is “*, le texte du deuxième alinéa*”.

EN–ZH	Sentences	W/S (EN)	W/S (ZH)
Train	84,871	13.5	13.8
Dev	734	14.3	14.5
Test	943	17.4	17.4

EN–FR	Sentences	W/S (EN)	W/S (FR)
Train	751,548	26.9	29.3
Dev	2,665	26.8	29.2
Test	2,655	27.1	29.4

Table 1: Summary of English–Chinese (EN–ZH) and English–French (EN–FR) datasets

3 Experiment

In our experiments, a baseline system **PB** is built with the phrase-based model in Moses (Koehn et al., 2007). We compare our approach with three other combination methods. **ADD** combines PB with *addition* (Ma et al., 2011), while **SUB** combines PB with *subtraction* (Koehn and Senellart, 2010). **WANG** combines SMT and TM at phrase-level during decoding (Wang et al., 2013; Li et al., 2014). For each phrase pair applied to translate an input phrase, WANG finds its corresponding phrase pairs in a TM instance and then extracts features which are directly added to the log-linear framework (Och and Ney, 2002) as sparse features. We build three systems based on our approach: **CWL_{add}** only uses constraints from *addition*; **CWL_{sub}** only uses constraints from *subtraction*; **CWL_{both}** uses constraints from both.

Table 1 shows a summary of our datasets. The EN–ZH dataset is a translation memory from Symantec. Our EN–FR dataset is from the publicly available JRC-Acquis corpus.⁴ Word alignment is performed by GIZA++ (Och and Ney, 2003) with heuristic function *grow-diag-final-and*.

⁴<http://ipsc.jrc.ec.europa.eu/index.php?id=198>

Systems	EN-ZH		EN-FR	
	BLEU \uparrow	TER \downarrow	BLEU \uparrow	TER \downarrow
PB	44.3	40.0	65.7	25.9
Sentence-Level Combination				
ADD	45.6*	39.2*	64.2	27.2
SUB	49.4*	36.3*	64.2	27.3
Phrase-Level Combination				
WANG	44.7*	39.3*	66.1*	25.7*
CWL _{add}	49.8*	35.7*	68.1*	23.7*
CWL _{sub}	51.4*	33.7*	68.6*	23.4*
CWL _{both}	51.2*	33.8*	68.3*	23.6*

Table 2: Experimental results of comparing our approach (CWL_x) with previous work. All scores reported are an average of 3 runs. Scores with * are significantly better than that of the baseline PB at $p < 0.01$. **Bold** scores are significantly better than that of all previous work at $p < 0.01$.

We use SRILM (Stolcke, 2002) to train a 5-gram language model on the target side of our training data with modified Kneser-Ney discounting (Chen and Goodman, 1996). Batch MIRA (Cherry and Foster, 2012) is used to tune weights. Case-insensitive BLEU [%] and TER [%] are used to evaluate translation results.

3.1 Results

Table 2 shows experimental results on EN-ZH and EN-FR. We find that our method (CWL_x) significantly improves the baseline system PB on EN-ZH by up to +5.5% BLEU score and by +2.4% BLEU score on EN-FR. In terms of TER, our system significantly decreases the error by up to -4.3%/-2.2% on EN-ZH and EN-FR, respectively.

Although, compared to the baseline PB, ADD and SUB work well on EN-ZH, they reduce the translation quality on EN-FR. By contrast, their phrase-level counterparts (CWL_{add} and CWL_{sub}) bring consistent improvements over the baseline on both language pairs. This suggests that a combination approach based on constrained word lattices is more effective and robust than sentence-level constrained translation. Compared to system WANG, our method produces significantly better translations as well. In addition, our approach is simpler and easier to adopt than WANG.

Compared with CWL_{add}, CWL_{sub} produces better translations. This may suggest that, for a constrained word lattice, *subtraction* generates a better sequence of constraints than *addition* since it keeps target words and the word order. However,

Ranges	Sentence	W/S (EN)
[0.8, 1.0)	198	16.4
[0.6, 0.8)	195	14.7
[0.4, 0.6)	318	16.8
(0.0, 0.4)	223	21.5

(a) English-Chinese

Ranges	Sentences	W/S (EN)
[0.9, 1.0)	313	32.5
[0.8, 0.9)	258	28.3
[0.7, 0.8)	216	28.4
[0.6, 0.7)	156	33.3
[0.5, 0.6)	171	34.1
[0.4, 0.5)	168	34.3
[0.3, 0.4)	277	40.3
(0.0, 0.3)	360	54.7

(b) English-French

Table 3: Composition of test subsets based on fuzzy match scores on English-Chinese and English-French data.

combining them together (i.e. CWL_{both}) does not bring a further improvement. We assume the reason for this is that *addition* and *subtraction* share parts of the constraints generated from the same TM. For example, in Figure 2, the edge “1 \rightarrow 5” based on *addition* and the edge “11 \rightarrow 7” based on *subtraction* are labeled by the same constraint.

3.2 Influence of Fuzzy Match Scores

Since a fuzzy match scorer⁵ is used to select the best TM instance for an input and thus is an important factor for combining SMT and TM, it is interesting to know what impact it has on the translation quality of various approaches. Table 3 shows statistics of each test subset on EN-ZH and EN-FR where sentences are grouped by their fuzzy match scores.

Figure 3 shows BLEU scores of systems evaluated on these subsets. We find that BLEU scores increasingly grow when match scores become higher. While ADD achieves better BLEU scores than SUB on lower fuzzy ranges, SUB performs better than ADD on higher fuzzy scores. In addition, our approaches (CWL_x) are better than the baseline on all ranges but show much more improvement on ranges with higher fuzzy scores.

⁵In this paper, we use a lexical fuzzy match score (Koehn and Senellart, 2010) based on Levenshtein distance to find the best match.

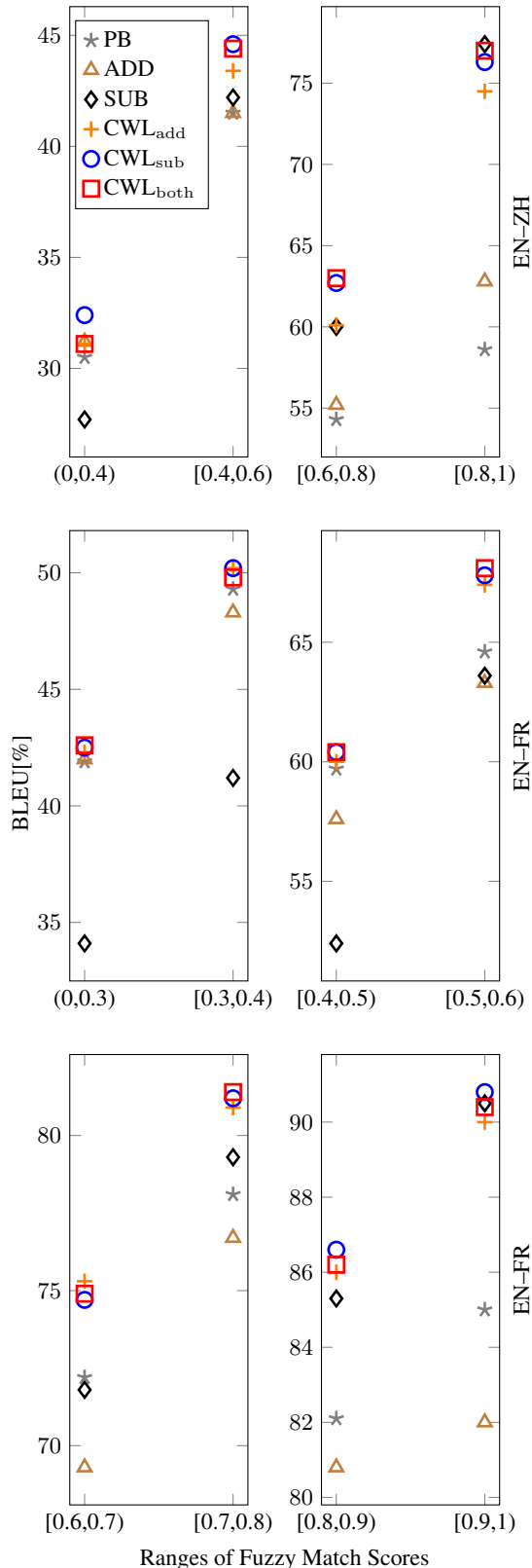


Figure 3: BLEU scores of systems evaluated on sentences which fall into different ranges according to fuzzy match scores on EN-ZH and EN-FR. All scores are averaged over 3 runs.

4 Conclusion

In this paper, we propose a *constrained word lattice* to combine SMT and TM at phrase-level. This method uses a word lattice to encode all possible phrasal constraints together. These constraints come from two sentence-level constrained approaches, including *addition* and *subtraction*. Experiments on English–Chinese and English–French show that compared with previous combination methods, our approach produces significantly better translation results.

In the future, we would like to consider generating constraints from more than one fuzzy match and using fuzzy match scores or a more sophisticated function to weight constraints. It would also be interesting to know if our method will work better when discarding fuzzy matches with very low scores.

Acknowledgments

This research has received funding from the People Programme (Marie Curie Actions) of the European Union’s Framework Programme (FP7/2007-2013) under REA grant agreement n° 317471. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund. The authors thank all anonymous reviewers for their insightful comments and suggestions.

References

- Ergun Biçici and Marc Dymetman. 2008. Dynamic Translation Memory: Using Statistical Machine Translation to Improve Translation Memory Fuzzy Matches. In *Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 454–465, Haifa, Israel, February.
- Stanley F. Chen and Joshua Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, pages 310–318, Santa Cruz, California, June.
- Colin Cherry and George Foster. 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, June.

- Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing Word Lattice Translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Columbus, Ohio, June.
- Yifan He, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging SMT and TM with Translation Recommendation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 622–630, Uppsala, Sweden, July.
- Philipp Koehn and Jean Senellart. 2010. Convergence of Translation Memory and Statistical Machine Translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31, Denver, Colorado, USA, November.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 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.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 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, June.
- Vladimir Iosifovich Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.
- Liangyou Li, Andy Way, and Qun Liu. 2014. A Discriminative Framework of Integrating Translation Memory Features into SMT. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas, Vol. 1: MT Researchers Track*, pages 249–260, Vancouver, BC, Canada, October.
- Yanjun Ma, Yifan He, Andy Way, and Josef van Genabith. 2011. Consistent Translation using Discriminative Learning - A Translation Memory-Inspired Approach. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1239–1248, Portland, Oregon, USA, June.
- Franz Josef Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302, Philadelphia, Pennsylvania, July.
- Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, March.
- Franz Josef Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4):417–449, December.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, July.
- M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 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, August.
- Andreas Stolcke. 2002. SRILM—an Extensible Language Modeling Toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 257–286, Denver, Colorado, USA, November.
- Kun Wang, Chengqing Zong, and Keh-Yih Su. 2013. Integrating Translation Memory into Phrase-Based Machine Translation during Decoding. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–21, Sofia, Bulgaria, August.