Baltic J. Modern Computing, Vol. 4 (2016), No. 2, pp. 165–177

Combining Translation Memories and Syntax-Based SMT

Experiments with Real Industrial Data

Liangyou LI[1], Carla PARRA ESCARTÍN[2], Qun LIU[1]

[1] ADAPT Centre, School of Computing, Dublin City University, Ireland
[2] Hermes Traducciones, Madrid, Spain

{liangyouli,qliu}@computing.dcu.ie
carla.parra@hermestrans.com

Abstract. One major drawback of using Translation Memories (TMs) in phrase-based Machine Translation (MT) is that only continuous phrases are considered. In contrast, syntax-based MT allows phrasal discontinuity by learning translation rules containing non-terminals. In this paper, we combine a TM with syntax-based MT via sparse features. These features are extracted during decoding based on translation rules and their corresponding patterns in the TM. We have tested this approach by carrying out experiments on real English–Spanish industrial data. Our results show that these TM features significantly improve syntax-based MT. Our final system yields improvements of up to +3.1 BLEU, +1.6 METEOR, and -2.6 TER when compared with a state-of-the-art phrase-based MT system.

Keywords: machine translation, translation memory, syntax-based SMT

1 Introduction

A Translation Memory (TM) is a database which stores legacy translations. Translators use them in their work because TMs allow them to increase their productivity by retrieving past translations and help them to enhance terminology and style cohesion across projects. Given an input sentence, a TM provides the most similar source sentence in the database together with its target translation as the reference for post-editing. If the input sentence was already translated in the past, the translator does not necessarily post-edit it. In the case of similar sentences (called "fuzzy matches"), the Computer Assisted Translation tool highlights the differences between the input sentence and the one stored in the TM to enhance the post-editing task. Different coloring schemes are used to highlight changes and additions to the source text in the TM to help the translator spot quicker the post-edits needed. As TMs can help produce high quality and

consistent translations for repetitive materials, they are believed to be useful for Statistical Machine Translation (SMT).

The combination of TM and SMT (henceforth referred as "TM combination") has been explored in many ways and it has shown to improve translation quality. Unlike the well-known pipeline approaches (Koehn and Senellart, 2010; Ma et al., 2011), which use a TM combination at sentence-level, run-time TM combination (namely, combining the TM and SMT during decoding) can make a better use of the matched sub-sentences (Wang et al., 2013; Li et al., 2014a). Such run-time combination has been explored on Phrase-Based (PB) MT (Koehn et al., 2003). However, PBMT systems making use of TMs only take into consideration continuous segments and thus generalizations such as the translation of the English *call. . . off* into the Spanish *cancelar* cannot be learned.

In this paper, we explore the possibility of using a run-time TM combination on syntax-based MT. Syntax-based MT learns translation rules which can be easily extrapolated to new sentences by allowing non-terminals. In our approach, for each applied translation rule during decoding, we identify a corresponding pattern in the TM and then extract sparse features which are subsequently added to our system.

In our experiments, the TM combination is done on the hierarchical phrase-based (HPB) model (Chiang, 2005) and the dependency-to-string (D2S) model (Xie et al., 2011; Li et al., 2014b). The experimental results on real English–Spanish data[3] show that syntax-based models produce significantly better translations than phrase-based models. After adding the TM features, the syntax-based models are further significantly improved.

2 TMs in SMT

Combining TMs and SMT together has been explored in different ways in recent years. He et al. (2010a) presented a recommendation system which used a Support Vector Machine (Cortes and Vapnik, 1995) binary classifier to select a translation from the outputs of a TM and an SMT system. He et al. (2010b) extended this work by re-ranking the N-best list of SMT and TM outputs. Koehn and Senellart (2010) and Ma et al. (2011) used TMs in a pipeline manner. Firstly, they identified the matched part from the best match in the TM and merged their translation with the input. Then, they forced their phrase-based SMT system to translate the unmatched part of the input sentence. One major drawback of these methods is that they do not distinguish whether a match is good or not at phrase-level.

Wang et al. (2013) proposed an improved method by using TM information on phrases during decoding. This method extracts features from the TM and then uses pre-trained generative models to estimate one or more probabilities added to phrase-based systems. However, their work requires a rather complex process to obtain training instances for these pre-trained models. Li et al. (2014a) simplified this method by extracting sparse features and directly adding them to systems. In experiments, this simplified method was comparable to the one in Wang et al. (2013). However, in both works, features are designed for phrase-based models.

[3] Our data belongs to a translation company and is further described in Section 5.1.

3 Syntax-Based SMT

Typically, syntax-based decoders are based on the CYK algorithm (Kasami, 1965; Younger, 1967; Cocke and Schwartz, 1970). It searches for the best derivation $d^* = r_1 r_2 \cdots r_N$ among all possible derivations D, as in Equation (1),

$$d^* = \underset{d \in D}{\operatorname{argmax}} P(d) \tag{1}$$

where r_i are the translation rules. Translations are carried out bottom-up. For each span of an input sentence, the decoder finds rules to translate it. The translation of a large span can be obtained by combining translations from its sub-spans using the syntactic rules containing non-terminals.

In this paper, we use two syntax-based models for our experiments. One is the HPB model (Chiang, 2005) which is based on formal syntax. The other one is the D2S model (Xie et al., 2011; Li et al., 2014b) which is based on dependency structures generated by the Stanford parser[4].

3.1 Hierarchical Phrase-Based Translation

A hierarchical phrase is an extension of a phrase by allowing gaps where other hierarchical phrases are nested. The HPB model is formulated by a synchronous context free grammar (SCFG) where gaps are represented by a generic non-terminal symbol X. Rules in the HPB are in the following form:

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle,$$

where γ is a string over source terminal symbols and non-terminals, α is a string over target terminal symbols and non-terminals, and $\sim$ is a one-to-one mapping between non-terminals in γ and α. An example of a rule is as follows:

$$X \rightarrow \langle \text{Bolivia holds } X_1, \text{Bolivia sostiene } X_1 \rangle,$$

where the index on each non-terminal indicates the mappings. These rules can be automatically learned from parallel corpora based on word alignments.

3.2 Dependency-to-String Translation

In the D2S model, there are two kinds of rules. One is the head rule which specifies the translation of a source word. For example:

$$\text{holds} \rightarrow \text{sostiene}$$

The other one is the head-dependent (HD) rule which consists of three parts: the HD fragment[5] s of the source side, a target string t and a one-to-one mapping ϕ from variables in s to variables in t, as in:

$$s = (\underline{\text{Bolivia}}) \text{ holds } (x_1\text{:selection})$$
$$t = \text{Bolivia sostiene } x_1$$
$$\phi = \{x_1\text{:selection} \rightarrow x_1\}$$

[4] `http://nlp.stanford.edu/software/lex-parser.shtml`

[5] An HD fragment is composed of a head node and all of its dependents.

Algorithm 1: Procedure for extracting a translation pattern from a TM instance.

Data: A rule r for an input sentence I, a TM instance (S, T, A)
Result: A translation pattern R for r
let $[i, j]$ denote the span covered by r;
$\langle [i_k, j_k] \rangle, k = 1 \cdots n$ are n subspans covered by non-terminals in r;
for *each span* $[i_k, j_k]$ **do**
    find a corresponding TM source span $[i_k^s, j_k^s]$, according to string edits;
    find a TM target span $[i_k^t, j_k^t]$, according to word alignment A;
end
find corresponding TM source and target spans $[i^s, j^s]$ and $[i^t, j^t]$ for $[i, j]$;
s = words in span $[i^s, j^s]$ and replacing phrases covered by $\langle [i_k^s, j_k^s] \rangle$ with non-terminals;
t = words in span $[i^t, j^t]$ and replacing phrases covered by $\langle [i_k^t, j_k^t] \rangle$ with non-terminals;
$R = \langle s, t, a \rangle$, a indicates mappings between non-terminals in s and t;

where the underlined element denotes the leaf node. Variables in the Dep2Str model are constrained either by words (like x_1:selection) or Part-of-Speech tags (like x_1:NN).

4 TM Combination Method

Inspired by Li et al. (2014a), who directly add sparse features to the log-linear framework of SMT (Och and Ney, 2002) to combine a TM with the PB model, in this paper we extract sparse features for each applied rule during decoding and directly add them to our syntax-based SMT systems. These features can be jointly trained with other features to maximize translation quality measured by BLEU (Papineni et al., 2002).

Given an input sentence in our test set, our approach starts from retrieving the most similar sentence from a TM.[6] The similarity is measured by the so-called fuzzy match score. Concretely, we use the word-based string-edit distance in Equation (2) (Koehn and Senellart, 2010) to compute the fuzzy match score between the input sentence and the TM instance.

$$F = 1 - \frac{\text{edit\_distance}(input, tm\_source)}{\max(\mid input \mid, \mid tm\_source \mid)} \tag{2}$$

During the calculation of the fuzzy match score, we also obtain a sequence of operations, including insertion, match, substitution and deletion, which are useful for finding the TM correspondence of an input phrase.

4.1 Recognizing Patterns in TM

Instead of translating a unique continuous phrase, the rules in our system can contain non-terminals which cover previously translated phrases. Before extracting any features for a rule, we first identify its corresponding patterns in the TM. The identification procedure is illustrated in Algorithm 1.

[6] In our experiments, we use the training corpus of our SMT experiments as a TM.

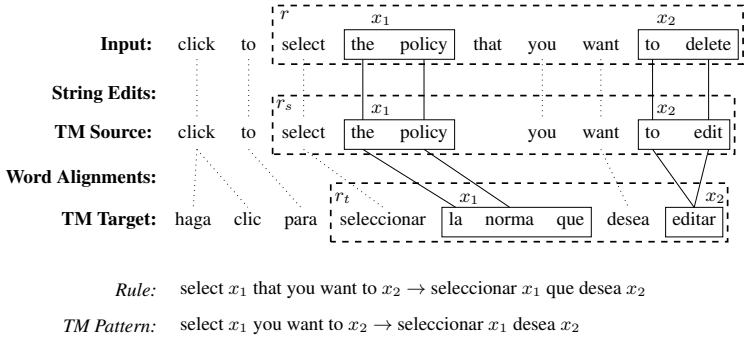

Fig. 1. An illustration of extracting translation patterns for a rule r. Phrases in solid rectangles are covered by non-terminals (x_i). Phrases in dashed rectangles are covered by translation rules or patterns.

For an input sentence I, we first retrieve an instance $\langle S, T, A\rangle$ from the TM, where S denotes the TM source segment, T is the TM target segment, and A indicates the word alignment between S and T. During decoding, rules are applied to translate I. Each rule r covers a continuous span $[i, j]$ of I and each non-terminal in r covers a sub-span of $[i, j]$ (lines 1–2 in Algorithm 1). For the span and its sub-spans, we first find their source correspondence in S according to the string edits between I and S and then the target correspondence in T according to the word alignment A (lines 3–7). Finally, we obtain a translation pattern $R = \langle s, t, a\rangle$ by replacing phrases covered by sub-spans with non-terminals (lines 8–10). Given a translation rule, Figure 1 illustrates how to identify a corresponding pattern in the TM.

Note that special cases might exist because of various situations in string edits and word alignments. Taking Figure 1 as an example, we cannot find a correspondence in the TM source for the input word *that*. In addition, the target t in extracted patterns can be extended by unaligned words, so in such case we might have multiple targets. These cases have been taken into consideration when extracting features (cf. Section 4.2).

4.2 Extracting Features

The features we use are similar to the ones in Li et al. (2014a) but modified to handle non-terminals in rules. Let $r = \langle \gamma, \alpha\rangle$ denote a rule we are using to translate an input sentence I. A retrieved TM instance for I is $\langle S, T, A\rangle$. The rule covers an input phrase s, which corresponds to TM segments $\langle s^r, t^r = (t^r_1 \cdots t^r_m)\rangle$. The following features are the same used by Li et al. (2014a):

- Feature $\mathbf{Z}_x$ ($x = 0 \cdots 10$) indicates the similarity between I and S. Each Z_x corresponds to a fuzzy match score range. For example, given a score $F(I, S) = 0.818$ which goes into the range [0.8,0.9), we obtain the feature Z_8.
- Feature $\mathbf{SEP}_x$ ($x = Y$ or N) is the indicator of whether s is a punctuation mark at the end of the input sentence I.

- Feature **NLN**$_{xy}$ ($x = 0, 1, 2$ and $y = 0, 1, 2$ and $y < x$) models the context of s^r and s, where x denotes the number of matched neighbors (left and right words) and y denotes how many of those neighbors are aligned to target words. If s^r is unavailable, we use feature NLN$_{non}$.
- Feature **CSS**$_x$ ($x = S, L, R, B$) describes the status of t^r. If t^r is unavailable, we use feature CSS$_{non}$. When $m = 1$ (i.e. the size of t^r is 1), $x = S$. $x = L, R, B$ means that t^r is obtained by extending unaligned words only on the left side or the right side or both sides, respectively.
- Feature **LTC**$_x$ ($x = O, L, R, B, M$) is the indicator of whether a t_i^r is the longest or not. If t^r is unavailable, we use feature LTC$_{non}$. $x = O$ means t_i^r is not generated by extending unaligned words. $x = L$ (or R, B) means t_i^r is only extended on its left (or right) side (or both sides) and has the longest left (or right) side (or both sides). $x = M$ means t_i^r is extended but not the longest one.

The assumed extracted translation patterns for r are $\langle \gamma^r, \alpha^r = (\alpha_1^r \cdots \alpha_m^r) \rangle$. We modify the following features and add them to our system:

- Feature **SPL**$_x$ measures the length of s, $x = 1 \cdots 7$ and *more*. Unlike PB models, where the phrase length is bounded, in syntax-based models we can use a rule to cover the whole input. So We use *more* to denote $\mid s \mid > 7$.
- Feature **SCM**$_x$ ($x = L, H, M$) represents the matching status between γ and γ^r, instead of s and s^r. This notation is used because γ might contain non-terminals, which in turn means that phrases covered by these non-terminals have already been considered. If γ^r is unavailable, we use feature SCM$_{non}$. Otherwise, L denotes a low similarity, namely $F(\gamma, \gamma^r) < 0.5$. H indicates $F(\gamma, \gamma^r) > 0.5$, and M means $F(\gamma, \gamma^r) = 0.5$.
- Similar to the SCM$_x$, feature **TCM**$_x$ ($x = L, H, M$) is the matching status between α and each α_i^r in α^r. If α^r is unavailable, we use feature TCM$_{non}$.
- We use the **CPM**$_x$ feature to model the reordering information. If γ only contains terminals, the feature is CPM$_{nnt}$. Otherwise, if α^r is unavailable, we use feature CPM$_{non}$. Otherwise, α^r and α define two permutations in terms of non-terminals in γ. The two permutations are assumed to be $p = p_1 \cdots p_n$ and $p^r = p_i^r \cdots p_n^r$. We use the Spearman correlation defined in Equation (3) to score the permutations.

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} (p_i - p_i^r)^2}{n(n^2 - 1)} \tag{3}$$

  The range of ρ is $[-1, 1]$. We divide the score into 5 groups, each of which indicates a feature: $x = nh$ when $\rho < -0.5$, $x = nl$ when $\rho \in [-0.5, 0)$, $x = 0$ when $\rho = 0$, $x = pl$ when $\rho \in (0, 0.5)$, and $x = ph$ when $\rho \geq 0.5$.

5 Experiments

5.1 Data

With the aim of further testing whether our experiments would be useful in a real commercial setting, we run our experiments on a real industrial data set. Our data belongs to

Table 1. Statistics of English–Spanish (EN–ES) corpus.

	Training	*Development*	*Test*
#sentences	577,639	1,959	1,964
#words (EN)	7,632,983	26,451	26,134
#words (ES)	9,049,260	31,170	31,195

a translation company and consists of all segments contained in the TM of one of their clients. The TM comprises all past projects of that client and is duly maintained and curated to ensure its quality. The data belongs to a technical domain and as mentioned earlier, it is used for English→Spanish translation tasks[7].

We deleted all repeated segments from the TM as well as all segments containing HTML tags occurring within the TM segments. While repetitions were deleted to follow the best practices in running SMT experiments, the segments with HTML tags were deleted because we found out that those segments were HTML addresses that did not require a translation and would have added noise to our data. Inline tags were not treated specifically and were maintained in the data. Once our data was cleaned, we randomly split it into *training*, *development* and *test*. Table 1 summarizes the size of our data in terms of number of sentences and running words.

5.2 Settings

In our experiments, we build four baselines. The two phrase-based baselines are: **PB**, the phrase-based model in Moses with default configurations, and **PBLR**, the phrase-based model, adding three lexical reordering models (Galley and Manning, 2008) to improve its reordering ability. The two syntax-based systems are: **HPB**, the hierarchical phrase-based model in Moses with default configurations, and **D2S**, an improved dependency-to-string model which has been implemented in Moses (Li et al., 2014b).[8] We add the TM features in Li et al. (2014a) to phrase-based systems and our TM features to syntax-based systems.

Word alignment is performed by GIZA++ (Och and Ney, 2004) with the heuristic function *grow-diag-final-and*. We use SRILM (Stolcke, 2002) to train a 5-gram language model on the target side of our training corpus with modified Kneser-Ney discounting (Chen and Goodman, 1996). Batch MIRA (Cherry and Foster, 2012) is used to tune weights. BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2011), and TER (Snover et al., 2006) are used for evaluation.[9]

5.3 Results and Discussion

Table 2 accounts for the results obtained for all our experiments. As may be observed, all our baselines are already pretty high and thus improvements are harder to obtain.

[7] Unfortunately, due to confidentiality agreements the data used in these experiments cannot be publicly released.

[8] http://computing.dcu.ie/ liangyouli/dep2str.zip

[9] `https://github.com/jhclark/multeval`

Table 2. Metric scores for all systems on English–Spanish. Each score is the average score over three MIRA runs (Clark et al., 2011). * means a system is better than PB at $p \leq 0.01$. $^{+}$ indicates a systems is better than PBLR at $p \leq 0.01$. **Bold** figures are significantly better than their no-TM counterparts at $p \leq 0.01$.

Systems	BLEU↑ (%)	METEOR↑ (%)	TER↓ (%)
PB	62.8	79.5	26.5
PBLR	63.5*	79.9*	26.0*
HPB	64.3^{+}	80.3^{+}	25.2^{+}
D2S	65.3^{+}	80.8^{+}	24.5^{+}
PB+TM	63.5*	79.9*	26.0*
PBLR+TM	64.2^{+}	80.3^{+}	25.5^{+}
HPB+TM	**65.9**	**81.0**	**24.3**
D2S+TM	**65.9**	**81.1**	**23.9**

This is not surprising, as we are working with an in-domain data set which is used for real translation tasks. The lexical reordering models significantly improve the PB system (+0.7 BLEU, +0.4 METEOR, and -0.5 TER). After incorporating the TM Combination approach (Li et al., 2014a), both systems (PB and PBLR) further produce significantly better translations. Both syntax-based systems (HPB and D2S) achieve significantly better results than phrase-based systems (up to +2.5 BLEU, +1.3 METEOR and -2.0 TER when comparing the PB system against the D2S system). Moreover, our TM features, when added to the HPB and D2S models, consistently improve both syntax-based baselines. In fact, these two systems (our final ones), achieve the best scores across all evaluation metrics (up to +3.1 BLEU, +1.6 METEOR, and -2.6). Example 1 shows how our D2S+TM and HPB+TM systems achieve better translations:

Example 1.

Source: button which opens the password entry window .
Ref: al pulsar este botón se abrirá la ventana de introducción de la contraseña .
TM Source: button which opens the container settings window .
TM Target: al pulsar este botón se abrirá la ventana configuración del repositorio .
TM Score: 0.75
PBLR: botón que abre la ventana de introducción de la contraseña .
HPB: botón que abre la ventana de introducción de la contraseña .
D2S: botón que abre la ventana de introducción de la contraseña .
PBLR+TM: botón que abre la ventana de introducción de la contraseña .
HPB+TM: al pulsar este botón se abrirá la ventana de introducción de la contraseña .
D2S+TM: al pulsar este botón se abrirá la ventana de introducción de la contraseña .

When using the TM Combination, both syntax-based models achieve a BLEU score of 1, while all other systems have a BLEU score of 0.6989. It shall be noted that both translations could actually be possible, but in our data there seems to be a stylistic preference: *se abrirá*, is preferred over *que abre*, which would be a more literal but still correct translation of the English "which opens". The TM Combination method allows our systems to learn the preferred translation in this case and match the reference. We have also found cases in which an error in the syntactic analysis causes our system to

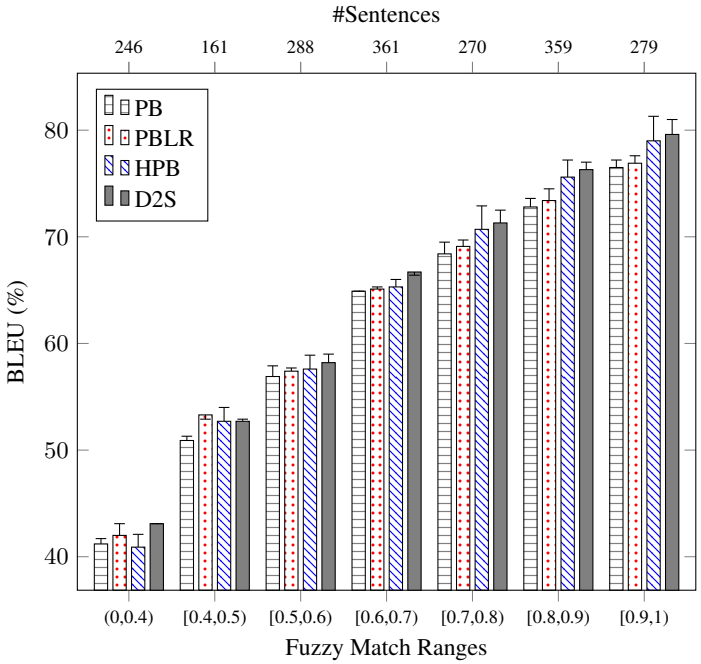

Fig. 2. BLEU scores evaluated on sentences grouped by fuzzy match scores. The symbol − on each bar indicates the BLEU scores of the systems after incorporating the TM approach.

fail (e.g. in the case of English nominal compounds), as well as cases in which our system produces a better translation than the reference.[10]

Since the fuzzy match score, as defined in Equation (2), is used to select a TM instance for an input sentence and thus is an important factor for combining the different SMT models and TM features, it is interesting to know the impact it has on the translation quality of the various systems we trained. Figure 2 shows BLEU scores of all systems evaluated on sentences grouped by fuzzy match scores. We first find that BLEU scores increase as fuzzy scores become higher. This is reasonable, since a higher fuzzy score means that we can find a similar sentence in the training data. The TM approach results in an improvement on almost all ranges. Such improvement is more consistent in the higher fuzzy ranges, namely [0.7,1). This also suggests that although TM instances with lower fuzzy scores could be useful, those with higher fuzzy scores are more reliable.

Another interesting finding is that D2S is consistently better than HPB. The main reason could be that rules in D2S are guided by linguistic annotations. In comparison with D2S, however, HPB benefits more when combined with the TM approach. In fact, when compared with their respective baselines (the same model without incorporating the TM approach), the HPB model is the one which experiences the highest improve-

[10] A qualitative analysis of our test set is being done to determine the real impact of our approach.

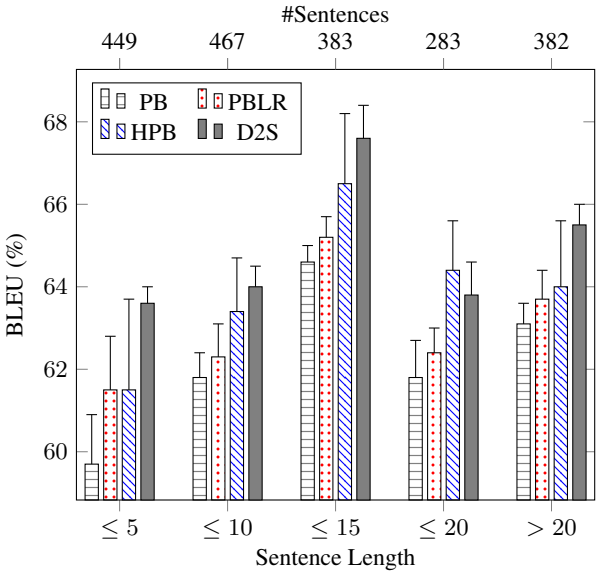

Fig. 3. BLEU scores evaluated on sentences grouped by sentence length. The symbol − on each bar indicates the BLEU scores of the systems after incorporating the TM approach.

ment (+1.6 BLEU, +0.7 METEOR, and -0.9 TER). This finding suggests that the TM approach could enhance the rule selection when linguistic annotations are unavailable, and that the TM approach achieves its greatest potential when combined with syntax-based models.

Finally, since syntax-based models learn translation rules which have a better generalization and reordering ability, we grouped sentences according to their length and evaluated our different systems by their respective sentence-length groups. The results for all systems are shown in Figure 3. As may be observed, the syntax-based systems, especially D2S, outperform the phrase-based models in all length ranges. Moreover, the TM combination consistently improves the results for all systems.

Bearing in mind that the ultimate goal would be to integrate an SMT system in a real commercial setting for MT Post-Editing tasks (MTPE), the results obtained suggest that the best option would be to use the D2S+TM system. Parra Escartín and Arcedillo (2015) investigated the productivity thresholds for MTPE tasks running an experiment with 10 professional translators in a real commercial setting. They found out that for English→Spanish MTPE tasks the productivity gain thresholds were of 45–50 BLEU and 25–30 TER. Given the results obtained by our systems, it seems that even the state-of-the-art baseline would already allow for a faster post-editing.

6 Conclusion

In this paper, we have explored how TM approaches can be used to enhance syntax-based SMT systems. To test our approach, we used real data from a translation company

and trained, tuned and tested different SMT systems on such data. The results of our experiments are very promising, particularly because improvements were achieved over already high baseline systems. The combination of the TM approach with other existing SMT systems yielded better overall scores (up to $+3.1$ BLEU, $+1.6$ METEOR, and -2.6 TER when compared with a state-of-the-art phrase-based MT system) in all MT evaluation metrics and for all sentence lengths.

Another interesting finding was that better BLEU scores seem to be obtained for the highest fuzzy match bands. In future research, we plan to run experiments of our TM Combination method taking only into consideration the higher fuzzies ([0.7,1)) to test whether better results are obtained and a threshold shall be established. We would also like to test our approach on public corpora and use a different data set from the TM to train SMT systems. It would be also interesting to know how effective each sparse feature is.

Acknowledgements

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^{o} 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. We also thank the anonymous reviewers for their insightful comments and suggestions.

References

Chen, S. F., Goodman, J. (1996). An Empirical Study of Smoothing Techniques for Language Modeling. *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, Santa Cruz, California, 310–318.

Cherry, C., Foster, G. (2012). Batch Tuning Strategies for Statistical Machine Translation. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Montreal, Canada, 427–436.

Chiang, D. (2005). A Hierarchical Phrase-based Model for Statistical Machine Translation. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor, Michigan, 263–270.

Clark, J. H., Dyer, C., Lavie, A., Smith, N. A. (2011). Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, Portland, Oregon, 176–181.

Cocke, J., Schwartz, J. T. (1970). Programming Languages and Their Compilers: Preliminary Notes. Technical report, Courant Institute of Mathematical Sciences, New York University, New York, NY.

Cortes, C., Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3):273–297.

Denkowski, M., Lavie, A. (2011). Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, 85–91.

Galley, M., Manning, C. D. (2008). A Simple and Effective Hierarchical Phrase Reordering Model. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, 848–856.

He, Y., Ma, Y., van Genabith, J., Way, A. (2010a). Bridging SMT and TM with Translation Recommendation. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 622–630.

He, Y., Ma, Y., Way, A., Van Genabith, J. (2010b). Integrating N-best SMT Outputs into a TM System. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, Beijing, China, 374–382.

Kasami, T. (1965). An Efficient Recognition and Syntax-Analysis Algorithm for Context-Free Languages. Technical report, Air Force Cambridge Research Lab, Bedford, MA.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, Prague, Czech Republic, 177–180.

Koehn, P., Och, F. J., Marcu, D. (2003). Statistical Phrase-based Translation. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, Edmonton, Canada, 48–54.

Koehn, P., Senellart, J. (2010). Convergence of Translation Memory and Statistical Machine Translation. *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, Denver, Colorado, USA, 21–31.

Li, L., Way, A., Liu, Q. (2014a). A Discriminative Framework of Integrating Translation Memory Features into SMT. *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas, Vol. 1: MT Researchers Track*, Vancouver, BC, Canada, 249–260.

Li, L., Xie, J., Way, A., Liu, Q. (2014b). Transformation and Decomposition for Efficiently Implementing and Improving Dependency-to-String Model In Moses. *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Ma, Y., He, Y., Way, A., van Genabith, J. (2011). Consistent Translation using Discriminative Learning - A Translation Memory-Inspired Approach. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, 1239–1248.

Och, F. J., Ney, H. (2002). Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, Pennsylvania, 295–302.

Och, F. J., Ney, H. (2004). The Alignment Template Approach to Statistical Machine Translation. *Compututational Linguistics*, 30(4):417–449.

Papineni, K., Roukos, S., Ward, T., Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, Pennsylvania, 311–318.

Parra Escartín, C., Arcedillo, M. (2015). Living on the edge: productivity gain thresholds in machine translation evaluation metrics. *Proceedings of the Fourth Workshop on Post-editing Technology and Practice*, Miami, Florida, 46–56.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. *Proceedings of Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA, 223–231.

Stolcke, A. (2002). SRILM-an Extensible Language Modeling Toolkit. *Proceedings of the 7th International Conference on Spoken Language Processing*, Denver, Colorado, USA, 257–286.

Wang, K., Zong, C., Su, K.-Y. (2013). Integrating Translation Memory into Phrase-Based Machine Translation during Decoding. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria, 11–21.

Xie, J., Mi, H., Liu, Q. (2011). A Novel Dependency-to-string Model for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Edinburgh, United Kingdom, 216–226.

Younger, D. H. (1967). Recognition and Parsing of Context-Free Languages in Time n^3. *Information and Control*, 10(2):189–208.

Received May 2, 2016 , accepted May 9, 2016