EXPERT

Project funded by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement no. 317471.

Project reference: 317471

Project full title: EXPloiting Empirical appRoaches to Translation

D6.1: Improved Components of Hybrid Systems

Authors: Liangyou Li (DCU)

Contributors: Jian Zhang (DCU), Qun Liu (DCU), Andy Way (DCU)

Document Number: EXPERT_D6.1_20150623

Distribution Level: Public

Contractual Date of Delivery: 31.10.14

Actual Date of Delivery: 23.06.15

Contributing to the Deliverable: WP6

WP Task Responsible: DCU

EC Project Officer: Concepcion Perez-Camaras

D6.1 Improved Components of Hybrid Systems

Liangyou Li
Supervisors: Prof. Qun Liu and Prof. Andy Way
School of Computing
Dublin City University
liangyouli@computing.dcu.ie

Contents

Abstract

Recent research suggests that hybrid systems are the future of translation. However, existing attempts to combine corpus-based translation approaches exploit only a limited number of ways to integrate them. In this report, we investigate how one of the corpus-based translation approaches, statistical machine translation, can be improved and how to easily integrate it with others, such as translation memory and example-based machine translation. The methods presented in this paper have a potential to be combined to produce a more robust hybrid system.

1 Introduction

In recent years, corpus-based machine translation has been explored in three directions: statistical machine translation (SMT), translation memory (TM) and example-based machine translation (EBMT). These three approaches translate sentences in different ways.

SMT automatically learns statistical models, such as a translation model and a language model (LM), to predict the probability of a target sentence being the translation of a given source sentence. The translation is produced by searching a target sentence, which has the highest score based on a weighted combination of these models. For training in SMT, translation pairs (parallel corpus) are needed to estimate model parameters. Given plenty of in-domain data, SMT can generate good quality translations. However, the quality drops dramatically for out-of-domain data. In addition, SMT results are fluent in short phrases but not good at large-size sentence structures, especially for distant languages.

Unlike with SMT, TM provides the most similar source sentence in the database together with the target translation as the reference to a human for post-editing. It has been extensively used to assist human translators. As TM stores legacy translations, it can produce high-quality and consistent translations for repetitive materials. However, it performs badly when there are no highly similar matches in TM.

Similar with TM, EBMT translates sentences by an analogy of existing translation examples. It does not need deep analysis on source texts and may generate high-quality translations when similar instances are found. However, rather than directly taking the target side of an example as a translation as in the TM, EBMT segments the input, translates each segment and recombines them together to generate a translation. The quality of EBMT is also dependent on the number of examples: typically the more the better. However, without learning and predicting, the coverage of examples is a major problem, especially for long sentences.

We can see that all of these three approaches have their own pros and cons. The work package 6 (WP6) of the EXPERT project is aimed at building a hybrid system which takes advantage of each translation approaches to produce better translations over each one of them.

Since SMT is more flexible to be extended, in this report we firstly investigate how SMT can be improved, which includes three parts. In the first part (Section 2), we conduct some experiments to highlight two problems of SMT: reordering and domain adaptation. These experiments bring us the confidence that SMT has a potential to be better. In the second part (Section 3), we incorporate linguistic information from dependency trees into SMT, which models long-distance relations between words in a sentence. In the third part (Section 4), we improve an SMT system via domain information, which shows how a large out-of-domain parallel corpus can be used to translate in-domain data. Furthermore, in Section 5, we integrate each individual translation approaches together via a less expensive way. Methods described in this report do not contradict with each other. Thus, a further combination of them will lead us to a more robust system.

2 Two Problems of SMT

To predict a translation, SMT breaks sentence pairs into smaller translation equivalence, either in word level, phrase level or syntax level. This results in a reordering problem. That is, SMT has to decide the order of translations of each segmentation. In addition, as stated in section 1, SMT also has problems on using and translating out-of-domain data. In this section, we conduct experiments to show that by alleviating these problems we can improve an SMT system significantly.

2.1 Baseline System

Our baseline system is based on the phrase-based model in Moses (Koehn et al., 2007) with default settings. A 5-order language model in our system is trained by SRILM (Stolcke, 2002) with Kneser-Ney discounting (Chen and Goodman, 1996). Alignment model is trained by MGIZA++ (Gao and Vogel, 2008) with *grow-diag-final-and* heuristic function. The system is tuned with k-best MIRA (Cherry and Foster, 2012) on a development set. We set the maximum iteration to be 25.

In our experiment, we use all provided German-English parallel data in WMT 2014. We apply language detection (Shuyo, 2010) for both monolingual and bilingual corpora to filter out Non-German and Non-English sentences. German compound words are splitted based on frequency (Koehn and Knight, 2003).

We take newstest 2013 as our test data and 2000 sentences from newstest 2008-2012 as our development data, which is selected by Feature Decay Algorithm (FDA) (Biçici and Yuret, 2014).

2.2 Reordering

Reordering in a German–English translation system is a considerable challenge since the two languages order words very differently. One approach to decrease the influence of word order is pre-reordering German words. However, such method is language-dependent. Another more general method is adopting separate reordering models. In our experiment, we use three Lexicalized Reordering Models (LRMs). They are word-based LRM (wLRM), phrase-based LRM (pLRM) and hierarchal LRM (hLRM) (Galley and Manning, 2008).

These three models have a different effect on the translation. Word-based and phrase-based LRMs focus on local reordering phenomenon, while hierarchical LRM can be applied into longer reordering. Table 1 shows the effectiveness of different LRMs. We can find that LRMs significantly improve the translation quality. When all three LRMs are adopted, our system achieves the best performance.

Systems	Tuning Set	newstest 2013
Baseline	–	24.2
+wLRM	23.8	25.1
+pLRM	23.9	25.2
+hLRM	**24.0**	**25.4**
+pLRM	23.8	25.1
+hLRM	23.7	25.2

Table 1: System BLEU[%] (Papineni et al., 2002) scores when different LRMs are adopted.

2.3 Domain Adaptation

Usually the size of in-domain data is small, while we can obtain a large amount of out-of-domain (or general domain) data without much effort. So we try to make use of all of them to build a more reliable system. In this section, we conduct experiments on language model (LM) interpolation.

In our baseline, LM is trained on all the monolingual data provided by WMT 2014. In this experiment, we build a larger LM by including data from the English Gigaword fifth edition (only taking partial data in the size of 1.6GB), the English side of the UN corpus and the English side of the 10^9 French–English corpus. Instead of training a single model on all data, we interpolate LMs trained on each subset by tuning weights to minimize the perplexity measured on the target side of the development set. In our experiment, after interpolation, the language model does not have a much lower perplexity. However, it significantly improves the system, as shown in Table 2.

Systems	Tuning Set	newstest 2013
Baseline	–	24.2
+LRMs	24.0	25.4
+LM Interpolation	**24.6**	**26.4**

Table 2: BLEU[%] scores on German–English corpus.

2.4 Conclusion

Of three popular corpus-based translation approaches, SMT has more potential to be generalized, thanks to its mathematical foundation. However, big problems also exist in SMT because it totally relies on learned statistical models after trained. In this section, we show that reducing the effect of two of these problems, reordering and domain adaptation, can significantly improve the translation quality.

3 Using Dependency Trees

One popular direction for addressing the reordering problem in SMT is to incorporate syntactic information. In this section, we use rules dependency structures to encode reordering patterns, since they provide grammatical relations between words, which have shown to be effective in SMT, especially for long distance reordering.

For instance, Shen et al. (2010) present a string-to-dependency model by using dependency fragments of neighboring words on the target side, which makes it easier to integrate a dependency language model. However, such string-to-tree systems run slowly (Huang et al., 2006). Menezes and Quirk (2005) and Quirk et al. (2005) propose a treelet (arbitrary connected subgraph) approach and use dependency structures on the source side. Xiong et al. (2007) extend the treelet approach to allow dependency fragments with gaps. However, these methods need another heuristic or separate reordering model to decide the best target position of the inserted words.

This work is based on a dependency-to-string (Dep2Str) model (Xie et al., 2011). It specifies reordering information in its rules and can perform a fast translation. For easily implementing this model, we transform an input dependency tree into a corresponding constituent tree. Then, we enrich this model via decomposing dependency structures. Table 3 shows a glossary for examples used in this section.

Chinese	English
Boliweiya	Bolivia
Juxing	holds
Zongtong	presidential
Yu	and
Guohui	parliament
Xuanju	elections

Table 3: A Chinese-to-English glossary.

3.1 Dependency-to-String Model

In the Dep2Str model, a head-dependent (HD) fragment, which is composed of a head node and all of its dependents, is the basic unit. Two kinds of rules are used. One is head rules which translate a source word. The other one is HD rules which consist of three parts: the HD fragment s of the source side, a target string t and a one-to-one mapping from variables in s to variables in t.

Figure 1 shows a derivation for translating a Chinese sentence into an English string in this model. The derivation proceeds from top to bottom. Variables in the higher-level HD rules are substituted by the translations of lower HD rules recursively.

3.2 Transformation of Dependency Trees

For easily implementing the Dep2Str model in the popular framework Moses, we transform a dependency tree into a corresponding constituent tree, where the source words are leaf nodes and all non-leaf nodes covering a phrase are labelled with categories which are usually variables defined in a tree-based model.

Accordingly, in the Dep2Str model, each variable represents a word (for head and leaf nodes) or a sequence of continuous words (for internal nodes). Thus, we use these variables to label non-leaf nodes of the produced constituent tree. Furthermore, the created nodes are constrained by the dependency information in the HD fragment.

Taking the dependency tree in Figure 1 as an example, its transformation result is shown in Figure 2.

3.3 Decomposition of Dependency Structures

The Dep2Str model treats a whole HD fragment as the basic unit, which may result in a data-sparsity problem. Thus, inspired by the treelet approach, we define that each HD fragment is decomposed into two smaller parts. This decomposition can be formulated as Equation (1):

$$\begin{aligned} & L_i \cdots L_1 H R_1 \cdots R_j \\ & \quad = L_m \cdots L_1 H R_1 \cdots R_n \\ & \quad + L_i \cdots L_{m+1} H R_{n+1} \cdots R_j \\ & \text{subject to} \\ & \quad i \geq 0, j \geq 0 \\ & \quad i \geq m \geq 0, j \geq n \geq 0 \\ & \quad i + j > m + n > 0 \end{aligned} \tag{1}$$

where H denotes the head node, L_i denotes the ith left dependent and R_j denotes the jth right dependent. Figure 3 shows an example.

We take advantage of this decomposition in two ways:

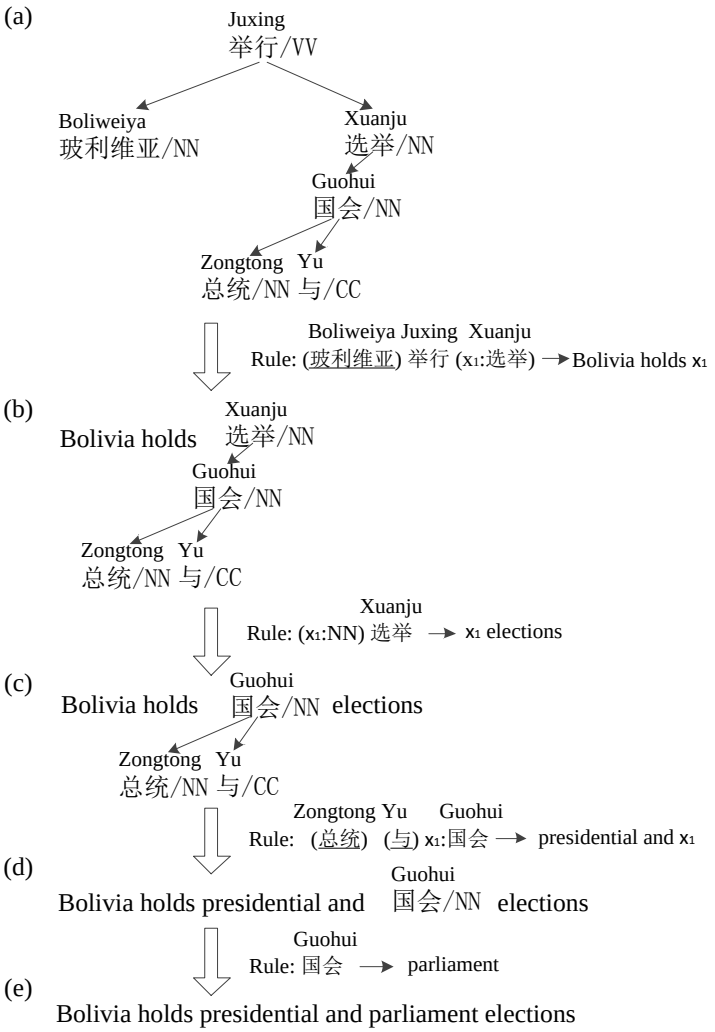

Figure 1: An example of a derivation for translating a Chinese dependency tree into an English string with rules in the Dep2Str model. Each rule translates a source word or a head-dependent fragment. Underlined elements indicate leaf nodes.

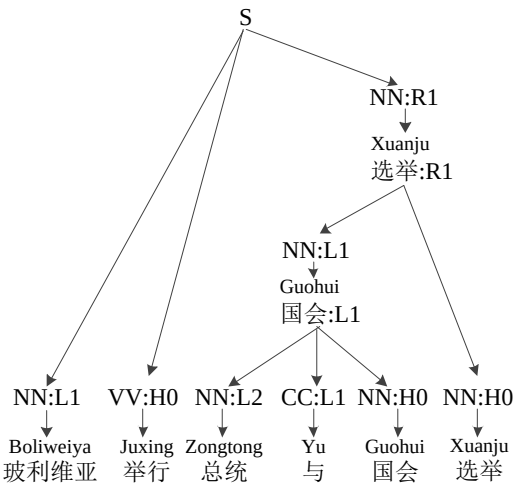

Figure 2: The corresponding constituent tree after transforming the dependency tree in Figure 1.

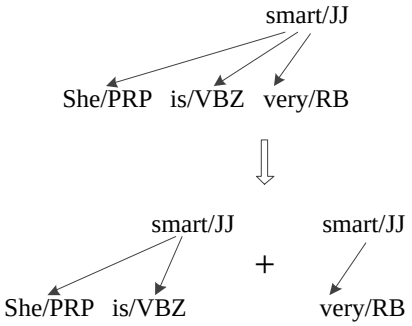

Figure 3: An example of decomposition on a head-dependent fragment.

- **Sub-structural Rules**

  We extract sub-structural rules by treating each possible sub-fragment as a new HD fragment, which are directly in the model.

- **Pseudo-Forest**

  For an HD fragment in the input dependency tree, we can translate one of its sub-fragments first, then obtain the whole translation by combining with translations of another sub-fragment, as shown in Figure 4. We encode the decomposition into the input dependency tree which results in a pseudo-forest. Figure 5 shows an example.

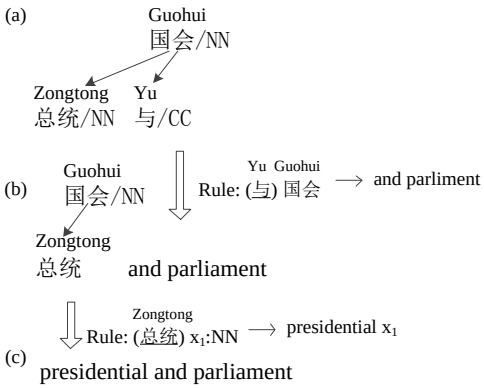

Figure 4: An example of translating a large HD fragment with the help of translations of its decomposed fragments.

3.4 Experiments

We conduct experiments on Chinese–English and German–English corpora.

3.4.1 Datasets and Settings

The Chinese–English training corpus is from the LDC data, including LDC2002E18, LDC2003E07, LDC2003E14, LDC2004T07, the Hansards portion of LDC2004T08 and LDC2005T06. We take NIST 2002 as the development set to tune weights, and NIST 2004 (MT04) and NIST 2005 (MT05) as the test data to evaluate the systems. Table 4 provides a summary of the Chinese–English corpus.

The German–English training corpus is from WMT 2014, including Europarl V7 and News Commentary. News-test 2011 is taken as the development set, while News-test 2012 (test12) and

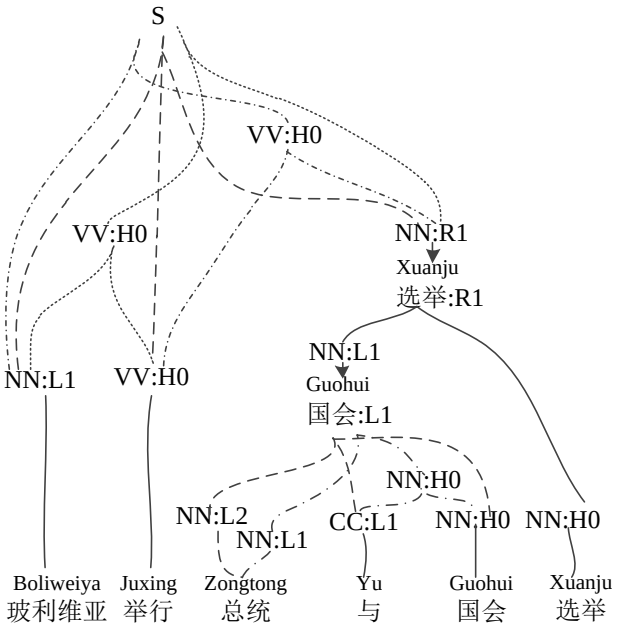

Figure 5: An example of a pseudo-forest. Edges drawn in the same type of line are owned by the same sub-tree. Solid lines are shared edges.

corpus	sentences	words(ZH)	words(EN)
train	1,501,652	38,388,118	44,901,788
dev	878	22,655	26,905
MT04	1,597	43,719	52,705
MT05	1,082	29,880	35,326

Table 4: Chinese–English corpus. For the English dev and test sets, words counts are averaged across 4 references.

corpus	sentences	words(DE)	words(EN)
train	2,037,209	52,671,991	55,023,999
dev	3,003	72,661	74,753
test12	3,003	72,603	72,988
test13	3,000	63,412	64,810

Table 5: German–English corpus. In the dev and test sets, there is only one English reference for each German sentence.

News-test 2013 (test13) are our test sets. Table 5 provides a summary of the German–English corpus.

Word alignment is performed by GIZA++ (Och and Ney, 2003) with the heuristic function *grow-diag-final-and*. We use SRILM to train a 5-gram language model on the Xinhua portion of the English Gigaword corpus 5th edition with modified Kneser-Ney discounting. Minimum Error Rate Training (Och, 2003) is used to tune weights. Case-insensitive BLEU is used to evaluate the translation results. Bootstrap resampling (Koehn, 2004) is also performed to compute statistical significance with 1000 iterations.

The hierarchical phrase-based model in Moses is employed for comparison in both language pairs. We take the default settings in Moses. This system is denoted as **HPB**. We denote our baseline Dep2Str model in Moses as **D2S**. The first experiment we do is to sanity check our implementation. We take a separate system (denoted as **XJ**) for comparison which implements the Dep2Str model based on Xie et al. (2011). The result is shown in Table 6. We find that using the transformation of dependency trees, the Dep2Str model implemented in Moses (D2S) is comparable with the standard implementation (XJ).

Systems	MT05
XJ	33.91
D2S	33.79

Table 6: BLEU score [%] of the Dep2Str model before (**XJ**) and after (**D2S**) dependency trees being transformed. Systems are trained on a selected 1.2M Chinese–English corpus.

The Dep2Str model only extracts phrase rules for translating a source word (head rules). This model can be enhanced by including phrase rules that cover more than one source word. Thus, we also conduct experiments where phrase pairs[1] are added into our system. We set the length limit for phrase 7.

3.4.2 Results

Chinese–English

In the Chinese–English translation task, Stanford Chinese word segmenter (Chang et al., 2008) is used to segment Chinese sentences into words. Stanford dependency parser (Chang et al., 2009) parses a Chinese sentence into the projective dependency tree.

Table 7 shows the translation results. We find that the decomposition approach, including sub-structural rules and pseudo-forest, improves the baseline system D2S significantly (absolute improvement of +1.53/+1.57). As a result, our system achieves comparable (-0.1/+0.14) results with the hierarchical phrase-based model (Chiang, 2005) in Moses. After including phrasal rules, our system performs significantly better (absolute improvement of +1.2/+0.68) than Moses HPB on both test sets.

German–English

We tokenize German sentences with scripts in Moses and use mate-tools[2] to perform morphological analysis and parse the sentence (Bohnet, 2010). Then MaltParser[3] converts the parse result into projective dependency trees (Nivre and Nilsson, 2005).

[1] In our experiments, the use of phrasal rules is similar to that of the HPB model, so they can be handled by Moses directly.

[2] `http://code.google.com/p/mate-tools/`

[3] `http://www.maltparser.org/`

Systems	MT04	MT05
Moses HPB	35.56	33.99
D2S	33.93	32.56
+pseudo-forest	**34.28**	**34.10**
+sub-structural rules	**34.78**	**33.63**
+pseudo-forest	**35.46**	**34.13**
+phrase	**36.76***	**34.67***

Table 7: BLEU score [%] of our method and Moses HPB on the Chinese–English task. We use bold font to indicate that the result of our method is significantly better than D2S at $p \leq 0.01$ level, and * to indicate the result is significantly better than Moses HPB at $p \leq 0.01$ level.

Systems	test12	test13
Moses HPB	20.44	22.77
D2S	20.05	22.13
+pseudo-forest	19.98	21.68
+sub-structural rules	**20.52**	**22.76**
+phrase	**20.91***	**23.46***
+pseudo-forest	20.25	22.24
+phrase	**20.75***	**23.20***

Table 8: BLEU score [%] of our method and Moses HPB on German–English task. We use bold font to indicate that the result of our method is significantly better than baseline D2S at $p \leq 0.01$ level, and * to indicate the result is significantly better than Moses HPB at $p \leq 0.01$ level.

Experimental results in Table 8 show that incorporating sub-structural rules improves the baseline D2S system significantly (absolute improvement of +0.47/+0.63), and achieves a slightly better (+0.08) result on test12 than Moses HPB. However, in the German–English task, the pseudo-forest produces a negative effect on the baseline system (-0.07/-0.45), despite the fact that our system combining both methods together is still better (+0.2/+0.11) than the baseline D2S. In the end, by resorting to phrasal rules, our system achieves the best performance which is significantly better (absolute improvement of +0.47/+0.59) than Moses HPB.

3.4.3 Discussion

Besides long-distance reordering (Xie et al., 2011), another advantage of the Dep2Str model is its simplicity. It can perform fast translation with fewer rules than HPB. Table 9 shows the number of rules in each system. It is easy to see that all of our systems use fewer rules than HPB. However, the number of rules is not proportional to translation quality, as shown in Tables 7 and 8.

Systems	# Rules	
	Zh-EN	DE-EN
Moses HPB	388M	684M
D2S	27M	41M
+sub-structural rules	116M	121M
+phrase	215M	274M

Table 9: The number of rules in different systems On the Chinese–English and German–English corpus. Note that pseudo-forest (not listed) does not influence the number of rules.

Experiments on the Chinese–English corpus show that it is feasible to translate the dependency tree via transformation. Such a transformation causes the model to be easily integrated into Moses without making changes to the decoder, while at the same time producing comparable results with the standard implementation (shown in Table 6).

The decomposition approach also shows a positive effect on the baseline Dep2Str system. Especially, sub-structural rules significantly improve the Dep2Str model on both Chinese–English and German–English tasks. However, experiments show that the pseudo-forest significantly improves the D2S system on the Chinese–English data, while it causes translation quality to decline on the German–English data.

Since using the pseudo-forest in our system is aimed at translating larger HD fragments via splitting it into pieces, we hypothesize that when translating German sentences, the pseudo-forest approach more likely results in much worse rules being applied. This is probably due to the shorter Mean Dependency Distance (MDD) and freer word order of German sentences (Eppler, 2013).

3.5 Conclusion

In this section we show that dependency information can be used to improve the translation quality of SMT. A dependency-to-string system can be easily implemented by transforming a dependency tree into a corresponding constituent tree. A decoder has been implemented in the popular translation framework Moses. Furthermore, decomposing a dependency structure into smaller pieces can enrich the system with more rules and allows us to create a pseudo-forest as an input for decoding. The code for this work is now available on `http://computing.dcu.ie/~liangyouli/dep2str.zip`.

4 Probabilistic Domain Indicator

Another challenge which rises above others in SMT is that the translation performance decreases when there are dissimilarities between the training and the testing environments. This type of challenge is typically defined as "domain adaptation". The underlying reasons that caused domain adaptation challenging are many, but the obvious one is that the training of an SMT system is heavily data-dependent.

One useful technique to alleviate this problem is called binary-featured phrase table fill-up (Nakov, 2008; Bisazza et al., 2011), which use a binary feature to indicate which domain each phrase pair comes from. In this section, we substitute the coarse binary feature with a probabilistic domain-likeness feature, which is assigned by a Support Vector Machine (SVM) (Cortes and Vapnik, 1995). This probabilistic fill-up improves the system significantly in our experiments.

One concern of our method is that a phrase pair in a translation table can be extracted from a number of different sentence pairs depending on the alignment applied and the extraction heuristic used. Accordingly, those training sentence pairs will be estimated to different domain-likeness feature values by the machine-learning algorithm used. We define the following three simple heuristics to address this issue:

- **Min**: the feature value uses the minimum domain-likeness estimations from the extracted sentence pairs. The motivation for this is if a phrase pair is extracted from a sentence pair, which has a lot of evidence to be excluded from the target domain, such a phrase pair should not be classified as in-domain even if other strong in-domain indicators are present.
- **Arithmetic Mean**: use the arithmetic mean of all the domain-likeness estimations. There is no bias to any sentence pair since each will still be able to contribute the final feature value.

Corpus	Train	Tune	Test
News Commentary (nc07)	42,884	1,064	2,007
Europarl (ep07)	1,257,436	–	–
TED (ted11)	106,642	934	1,664
news-commentary-v9 (ncv9)	181,274	–	–

Table 10: SMT training corpus statistics

- **Geometric Mean**: use the geometric mean value to describe the central tendency of all domain-likeness estimations.

4.1 Feature Set

We take inspiration from the previous works in Axelrod et al. (2011) and design three sets of features for each SMT training sentence pair:

- **Source Domain Features**: the domain evidence shown from the source side of the training data. We use the perplexity value computed from the in- and general-domain language models in this feature set.
- **Target Domain Features**: the domain evidence shown from the target side of the training data. We use the perplexity value computed from the in- and general-domain language models in this feature set.
- **Domain Distance Features**: a feature set similar to the language model data-selection approach in Axelrod et al. (2011). We use both the source-side and target-side perplexity difference in this feature set.

4.2 Experiment

4.2.1 Datasets

The experiments use data from WMT07, WMT13 and IWLST11 translation tasks on French-to-English language pair, as summarized in Table 10.

There are two fill-up experiments designed to evaluate our approach, defined as *prob-fill-up_heuristic(in-domain,general-domain)*, such as *prob-fill-up_heuristic (nc07,ep07)* and *prob-fill-up_heuristic (ted11,ncv9)*. The experimental design is to assess our approach in both of the following situations: (i) general-domain dataset being significantly larger than the in-domain data, and (ii) the two datasets being similar in size.

4.2.2 SVM training

SVM training is a supervised learning process so having labeled training data available is essential. The label is either in-domain or general-domain for the SVM training instance in our case. The in-domain labeled SVM training data can be obtained directly from the SMT training set, but the general-domain data is mixed with in- and out-of-domain instances. We simply randomly select M number of general-domain and in-domain sentences as SVM training instances in our experiments.

The data used for SVM training, language model training (for extracting features) and SVM tuning are summarized in Table 11. The optimized precision of SVM is presented in Table 12.

Experiment	M	N	T
prob-fill-up(nc07,ep07)	42,884	40,000	2,884
prob-fill-up(ted11,ncv9)	50,000	45,000	5,000

Table 11: SVM data statistics, where M,N and T are the data sizes (in sentences) used for training, tuning and testing, respectively.

Experiment	Accuracy
prob-fill-up(nc07,ep07)	0.8139
prob-fill-up(ted11,ncv9)	0.8565

Table 12: Optimized SVM precision.

4.2.3 Translation System Training

All SMT systems in our experiments are trained using the phrase-based SMT in Moses. We use the word aligner MGIZA++ for word alignment in both translation directions, and then symmetrize the word alignment models using the heuristic of *grow-diag-final-and*. The translation systems are tuned with MERT. A 5-gram language model is trained with the IRSTLM toolkit (Federico et al., 2008) using all the available target sentences in each of the fill-up experiment scenarios.

4.2.4 Results

We set our baseline systems to be the fill-up system of Bisazza et al. (2011) (*fill-up(experiment)*), which has been integrated within the Moses framework. Tables 13 and 14 report our results using case-insensitive BLEU. We use † to indicate where the probabilistic feature-based fill-up approach systems (*prob-fill-up_heuristic(experiment)*) achieve significant improvement compared with the baseline system at the level $p = 0.01$ level with 1000 iterations.

System	Test (news-test2007)
fill-up(nc07,ep07)	28.01
prob-fill-up_Min(nc07,ep07)	28.03
prob-fill-up-Arithmetic_Mean(nc07,ep07)	28.21
prob-fill-up-Geometric_Mean(nc07,ep07)	28.37†

Table 13: prob-fill-up_heuristic(nc07,ep07) experiment BLEU scores on testing data, the significance testing at the level $p = 0.01$ level with 1000 iterations.

The result of the *prob-fill-up_heuristic(nc07,ep07)* experiment in Table 13 shows that the probabilistic feature-based fill-up systems using three heuristics for domain-likeness calculation can improve the translation performance over the baseline system. The system using the central tendency heuristic for the domain-likeness estimation outperforms the other, obtaining 0.36 absolute BLEU score and 1.3% relative improvement over the baseline system, and $p = 0.01$ significant improvement.

In our second experiment as shown in Table 14, the geometric mean calculation produces a strong BLEU score, +0.39 (1.3% relative) higher in contrast with the baseline system. However, the arithmetic mean calculation achieves the best result in this experiment with a 31.64 BLEU score (2.66% relative) on the test set. Both of the above two systems in our last experiment qualify as statistically significant improvements over the baseline system at $p = 0.01$ level.

System	Test (tst2011)
fill-up(ted11,ncv9)	30.82
prob-fill-up_Min(ted11,ncv9)	30.73
prob-fill-up_Arithmetic_Mean(ted11,ncv9)	31.64†
prob-fill-up_Geometric_Mean(ted11,ncv9)	31.21†

Table 14: prob-fill-up_heuristic(ted11,ncv9) experiment BLEU scores on testing data, the significance testing at the level $p = 0.01$ level with 1000 iterations.

4.2.5 Data selection

We compare our probabilistic feature-based fill-up with the data selection approach proposed in Axelrod et al. (2011). We first rank the general-domain corpus according to the sum of in- and out-of-domain perplexity difference normalized by the corresponding sentence length, defined as in (2), with the ranking in reverse order:

$$PPL - DIFF = \frac{[PPL_{I\_src(S)} - PPL_{O\_src(S)}]}{length(S)} + \frac{[PPL_{I\_tgt(T)} - PPL_{O\_tgt(T)}]}{length(T)}, \quad (2)$$

where S and T are the source and target sentences, respectively. The top p proportion of the ranked general-domain corpus is selected, and concatenated with the in-domain corpus to train the data selection systems.

Figures 6 and 7 illustrate the effects of the selection proportion on the BLEU score of SMT systems. As we might expect, additional general-domain training instances can benefit SMT performance, with 20% of *ep07* and 65% of *ncv9* selection, obtaining 27.28 and 31.73 BLEU scores, respectively. In addition, it is harmful to include a large proportion of general-domain data, which can overtake the in-domain data and cause target-domain bias. In contrast, the proposed probabilistic feature-based fill-up approach is able to efficiently use all the general-domain data, achieving significantly better translation results on the *(nc07,ep07)* dataset and comparable translation results (Table 5) on the *(ted11,ncv9)* dataset.

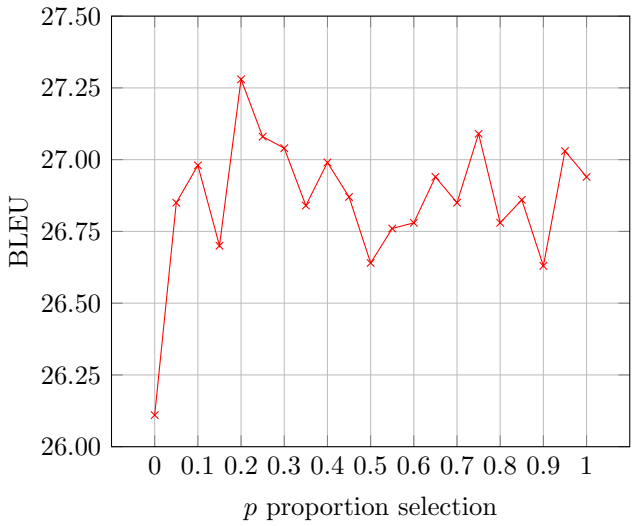

Figure 6: BLEU scores with different p proportion of data selection on (nc07,ep07) dataset.

4.2.6 Domain-likeness Distribution

Figure 8 compares for the interval grouped range between 0.10 to 1.00, the percentage of phrase entries contributing to the overall phrase table. It shows that the general-domain training

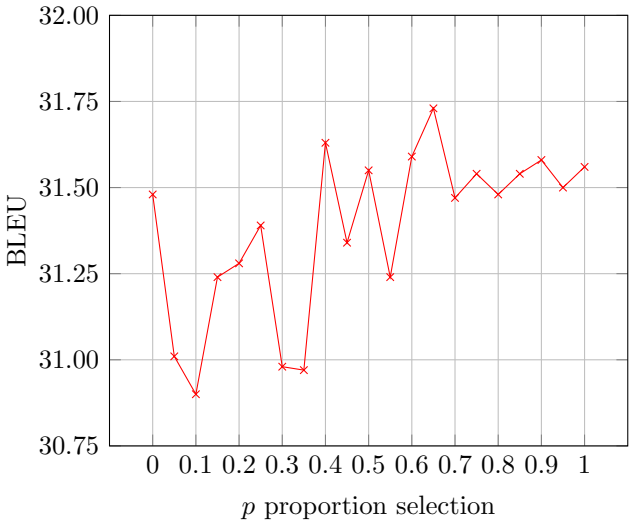

Figure 7: BLEU scores with different p proportion of data selection on (ted11,ncv9) dataset.

sentences can provide different levels of utility, and can be beneficial (in the case of probability feature value >0.5) or harmful (in the case of probability feature value <0.5) to the merged phrase table.

Haddow and Koehn (2012) also found that general-domain training data can benefit the translation table most when it is just allowed to add entries, but scores from the general-domain may be harmful to translation quality. Previous work tries to address this question by defining a fairness feature value to all phrase pairs extracted from the general-domain training sentences. However, such a fairness feature value may cause the potential in-domain phrase entries to be treated unjustly. Using a probabilistic feature value representing domain-likeness can distinguish between the extracted phrase pairs and also provides a soft-handed approach for phrase-table merging.

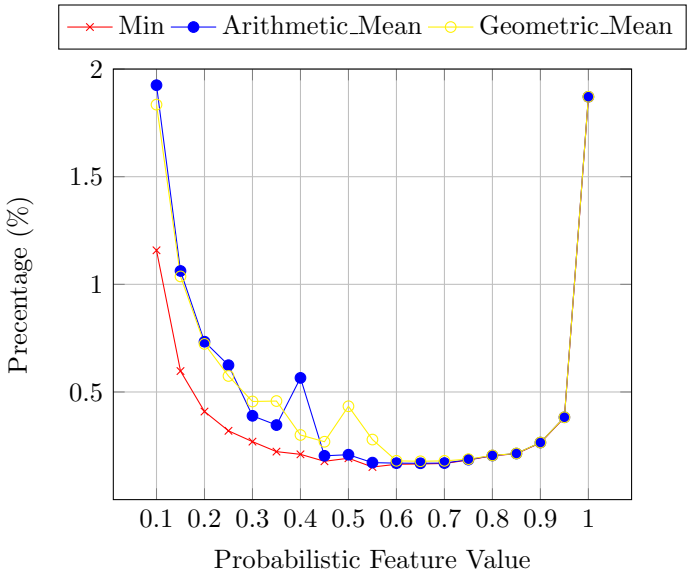

Figure 8: The distribution of *Min*,*Arithmetic_Mean* and *Geometric_Mean* phrase pairs contribution comparison: X-axis represents the range from 0.10 to 1.00. Y-axis represents the percentage of phrase entries to the overall testing data filtered phrase table.

4.3 Conclusion

We extended the fill-up phrase-table merging approach by assigning a domain-likeness probabilistic feature. Two experimental scenarios show that our fill-up approach is a soft-handed dynamic approach and can significantly improve translation performance in both experiments compared to previous fill-up studies. However, the approach shown in this section is still preliminary and can be extended further.

5 Integration in A Discriminative Framework

In this section, we integrate TM into SMT in a discriminative framework by incorporating TM-related feature functions. With the capacity of handling a large amount of features in the framework, we propose a method to efficiently use multiple fuzzy matches, which bring more feature functions.

In this section we only conduct experiments on SMT/TM combination. However, under the discriminative framework, EBMT-based features can also be easily added.

5.1 The Discriminative Framework

Generally, in a state-of-the-art statistical translation framework like Moses, a discriminative framework (Och and Ney, 2002) assigns the direct translation probability. When considering TM, this framework is generalized to Equation (3):

$$P(e \mid f, D) = \frac{\exp\{\sum_{m=1}^{M} \lambda_m h_m(e, f, D)\}}{\sum_{e'} \exp\{\sum_{m=1}^{M} \lambda_m h_m(e', f, D)\}}, \tag{3}$$

where D denotes instances (sentence pairs) from TM. Then, we obtain the rule in Equation (4):

$$\begin{aligned} e &= \underset{e'}{\operatorname{argmax}}\{P(e' \mid f, D)\} \\ &\simeq \underset{e'}{\operatorname{argmax}}\{P(e' \mid f, D_f)\} \\ &\simeq \underset{e'}{\operatorname{argmax}}\{\sum_{m=1}^{M} \lambda_m h_m(e', f, D_f)\} \end{aligned} \tag{4}$$

where h_m are feature functions, λ_m are weights.

In this paper, we change features defined in Wang et al. (2013) to TM feature functions and directly add them into a phrase-based system. In decoding, a foreign input sentence f is segmented into a sequence of I phrases $\overline{f}_1^I$ and each foreign phrase $\overline{f}_i$ is translated into a target phrase $\overline{e}_i$. Thus, a TM-related feature function can be seen as the sum of I feature functions which are based on phrase pairs, as in Equation (5):

$$\begin{aligned} h(e, f, D_f) &= h(\overline{e}_1^I, \overline{f}_1^I, D_{\overline{f}_1^I}) \\ &\simeq \sum_{i=1}^{I} h(\overline{e}_i, \overline{f}_i, D_{\overline{f}_1^I}) \end{aligned} \tag{5}$$

where $h(\overline{e}_i, \overline{f}_i, D_{\overline{f}_1^I})$ is measured on the phrase pair $(\overline{e}_i, \overline{f}_i)$ and TM matches $D_{\overline{f}_1^I}$.

5.2 Fuzzy Matching

In this paper, TM-related features are extracted from matches in the TM. For retrieving matches, we use a word-based string edit distance (Koehn and Senellart, 2010) to measure

the similarity between an input sentence and a TM instance, as in Equation (6):

$$FMS = 1 - \frac{\text{edi\_distance}(input, tm\_source)}{\max(\mid input \mid, \mid tm\_source \mid)} \quad (6)$$

During the calculation of the fuzzy match score, we also obtain a sequence of operations, including insertion, match, substitution and deletion, to convert the input sentence into a TM instance. Such operations are useful for finding the TM correspondence of a source phrase.

5.3 TM Features

Wang et al. (2013) propose a deep integration method by using TM information during decoding. They extract features from the best match in the TM and use pre-trained generative models to estimate one or more probabilities and then add them into a phrase-based system for scoring a translation. However, their work requires a rather complex process to obtain training instances for these pre-trained models and needs to define the generative relation between different features. In this paper, we avoid using pre-trained models and tune feature weights to directly maximize BLEU scores.

Given an input sentence and its best match in the TM, for each phrase pair applied to the input, we first find its corresponding TM source phrase based on operations for calculating edit-distance. Then, we identify one or more TM target phrases. Then we extract features for the phrase pair. These features are summarized as follows:

- the similarity between the best and the input;
- the similarity between the source phrase and TM source phrase;
- the length of the source phrase;
- an indicator of whether the source phrase is the punctuation at the end of the input or not;
- the similarity between the target phrase and TM target phrases;
- matching and alignment status in context between the source phrase and the TM source phrase;
- alignment status of TM target phrases;
- an indicator of whether a TM target phrase is the longest or not;
- reordering information.

5.4 Multiple Fuzzy Matches

In this paper, besides the best match, we also find a TM instance for each source phrase. We propose a method to find multiple matches to cover as many words in the input as possible: for each source phrase, we find a TM instance, which contains this phrase and has the highest fuzzy match score with the input. We call such a TM instance **span-match**. Figure 9 shows an example of finding span-matches. When we extract features for phrase 1, we use TM source 1 and its translation as the match. Similarly, for phrase 2, we use TM source 2; and for phrase 3, we find TM source 3 and use it for feature extraction.

Features from span-matches are similar to those from the best match. We distinguish features from the best match and span-matches. In addition, we also define two more features:

- Feature *NO_SPAN_MATCH* means we cannot find a span-match for the current source phrase.

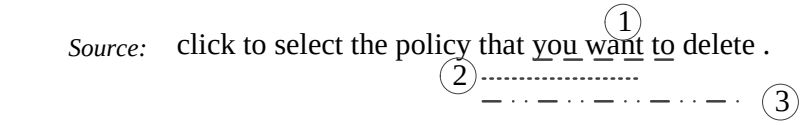

Figure 9: An example of finding multiple matches.

EN-ZH	sentences	words(EN)	words (ZH)
train	86,602	1,148,126	1,171,313
dev	762	10,599	10,791
test	943	16,366	16,375

EN-FR	sentences	words(EN)	words (FR)
train	765,922	20,604,865	22,401,839
dev	1,902	67,403	73,743
test	1,919	71,228	78,177

Table 15: Summary of English–Chinese (EN-ZH) and English–French (EN-FR) corpora

- Feature *IS_SPAN_BEST* means the span match is equal (the same fuzzy match score) to the best match.

Different to the best match which is estimated over the whole sentence and thus does not bias to any particular source phrase, a span-match provides us information about how a specific source phrase is used and thus may be helpful in selecting a proper target candidate. In addition, note that for a source sentence, the number of span-matches used is flexible, so our method does not need to set a threshold and do optimization on such a parameter.

5.5 Experiment

We conduct experiments on English-Chinese and English–French corpus.

5.5.1 Datasets

Our English-Chinese data set is a translation memory from Symantec, as shown in Table 15. Our English–French data is from the publicly available JRC-Acquis corpus.[4] We randomly select 3000 sentence pairs as dev data and 3000 as test data. We filter sentence pairs longer than 80 words in the training data and 100 words in the dev and test data. We also keep the length ratio less than or equal to 3 in all data sets. Table 15 shows a summary of English–French corpus.

5.5.2 Baseline

On both language-pairs, we take the phrase-based model in Moses with default settings as our baseline. Word alignment is performed by GIZA++, with heuristic function *grow-diag-final-and*. We use SRILM to train a 5-gram language model on the target side of the training data

[4] `http://ipsc.jrc.ec.europa.eu/index.php?id=198`

systems	EN–ZH		EN–FR	
	dev	test	dev	test
Phrase-based SMT	52.88	44.63	61.65	61.75
+Wang's model	**54.47**	**45.72**	**62.45**	**62.44**
+TM feature	**54.71**	**45.89**	**62.76**	**62.43**
+multiple fuzzy matches	**55.48***	**46.75***	**63.38***	**63.10***

Table 16: BLEU [%] on English–Chinese (EN-ZH) and English–French (EN-FR) data. Bold figures mean that the result is significantly better than the baseline phrase-based model at $p \leq 0.01$ level. * indicates that multiple fuzzy matches significantly improves the system with TM features at $p \leq 0.01$ level.

Ranges	Sentence	Words(EN)	Words/Sentence
[0.8, 1.0)	198	3,239	16.4
[0.6, 0.8)	195	2,876	14.7
[0.4, 0.6)	318	5,358	16.8
(0.0, 0.4)	223	4,784	21.5

(a) English–Chinese

Ranges	Sentence	Words(EN)	Words/Sentence
[0.9, 1.0)	313	10,166	32.5
[0.8, 0.9)	258	7,297	28.3
[0.7, 0.8)	216	6,128	28.4
[0.6, 0.7)	156	5,195	33.3
[0.5, 0.6)	171	5,832	34.1
[0.4, 0.5)	168	5,754	34.3
[0.3, 0.4)	277	11,157	40.3
(0.0, 0.3)	360	19,699	54.7

(b) English–French

Table 17: Composition of test subsets based on fuzzy match scores on English–Chinese and English–French data.

with modified Kneser-Ney discounting. MERT is used to tune weights. However, when TM features are incorporated, the number of features grows to more than 50. As MERT is known to be weak when the number of features grows (Durrani et al., 2013), we use MIRA instead to tune weights in this case. We set the maximum iteration of MIRA to be 25.

5.5.3 Results

Table 16 shows our experiment results on two language pairs. We found that our system with TM features achieves comparable results (+0.24/+0.31 on the dev set and +0.17/-0.01 on the test set) with Wang et al. (2013) and both systems are significantly better than the baseline. After multiple fuzzy matches are incorporated, our system shows further significant improvement (+0.76/+0.62 on dev and +0.86/+0.67 on test).

In addition, we are also interested in the performance of the systems on different fuzzy match ranges. Table 17 shows statistics on subsets of test data based on fuzzy match ranges on English–Chinese and English–French data. We see that sentences with a lower fuzzy match score (0.0-0.4) are longer.

The BLEU scores [%] for different fuzzy match ranges are shown in Figure 10. It is clear that our system with multiple fuzzy matches achieves the best performance over most ranges. Especially on the English–Chinese task, when both Wang's model and the TM features are

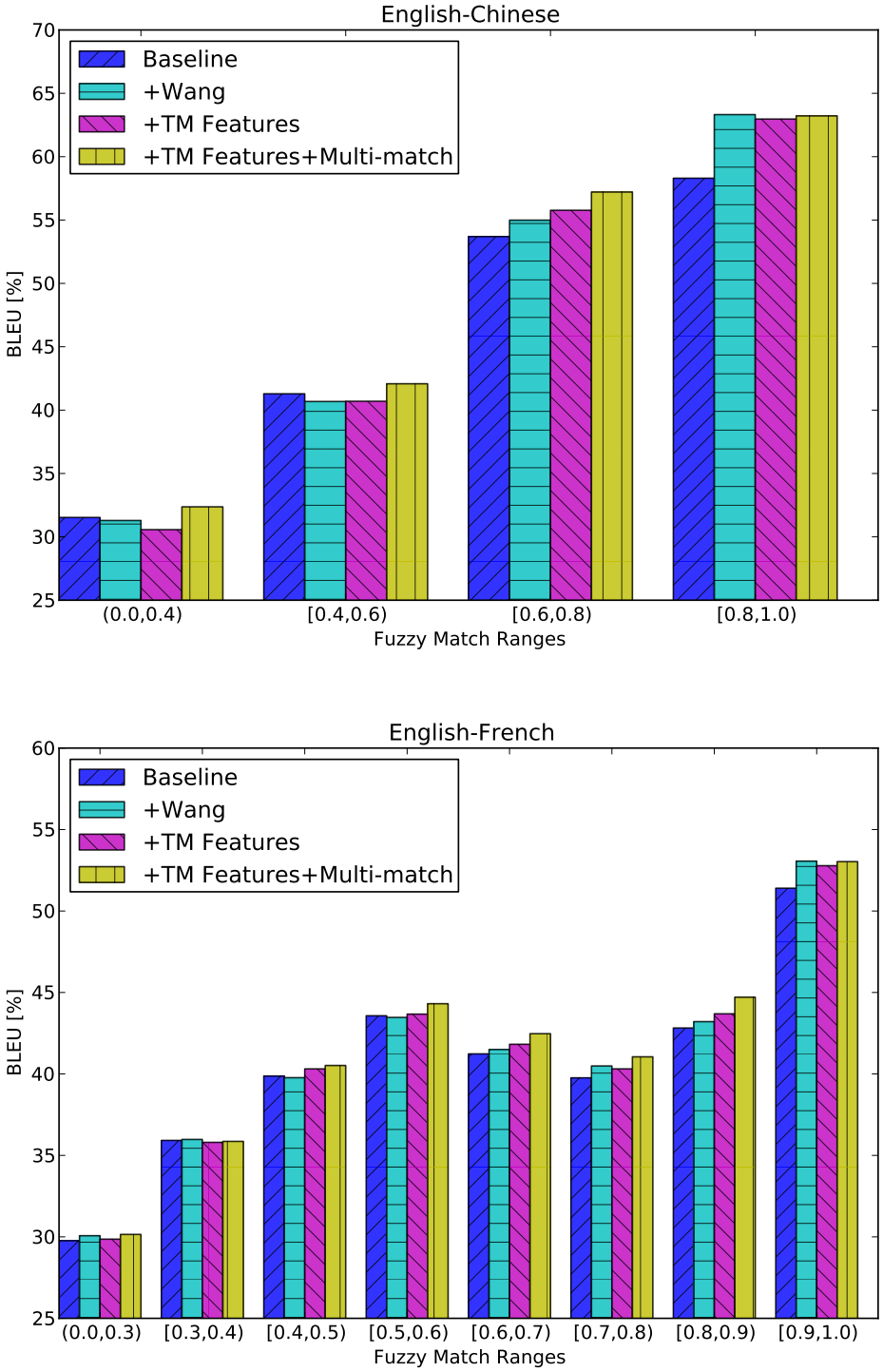

Figure 10: BLEU [%] for different fuzzy match ranges on two language pairs. The baseline is the phrase-based SMT system. The other three systems integrate different TM information into the baseline.

ineffective on the range (0.0,0.4) and [0.4,0.6), multiple fuzzy matches improve the system to give the best translation on both language pairs. However, in the highest range, Wang et al. (2013)'s method gives the best results. It suggests that our system does not bias to high-scoring fuzzy match range and treat all ranges fairly.

5.6 Conclusion

In this section, we present a discriminative framework which can integrate TM into SMT. Under this framework, we add TM feature functions, which model the relation between the source sentence and TM instances, into a phrase-based SMT. In experiments, our method performs significantly better than the baseline phrase-based system. Furthermore, we present a method to efficiently use multiple fuzzy matches. Experiments show that this addition significantly improves our system.

Although features used currently are from Wang et al. (2013), this method is much simpler yet shows comparable results to their work. In addition, this method can be more easily extended with additional features and integrate EBMT into the system.

6 Conclusions

Suggested by recent research, corpus-based translation approaches, including statistical machine translation (SMT), translation memory (TM) and example-based machine translation (EBMT), have their own pros and cons. The combination of them, resulting in a hybrid system, can improve the quality of translations.

With a solid mathematical foundation, SMT learns models during training and translates new input sentences by making a prediction, which is independent of the training data. So SMT can be easily generalized. However, big challenges still exist in SMT. As pointed in this report, dealing with the reordering problem and domain adaptation can be useful to improve the translation quality of SMT.

To obtaining a better-reordered translation, we choose to incorporate dependency trees on the source side into SMT. Based on a dependency-to-string system, we show that an easy implementation of such system is possible by transforming dependency trees into its constituent counterparts. We also present a decomposition method that dis-constructs a large dependency structure into smaller parts. Then we use these sub-structures to enrich the model with more rules and translate large structures.

To better using out-of-domain (or general domain) corpus to translate in-domain data, we show a probabilistic phrase-table fill-up method, which assigns a probability to each entry in the phrase table. This probability indicates the likeliness of the entry being in domain, which distinguishes phrase pairs from different domains and provides a soft-handed approach for phrase-table merging.

By generalizing the discriminative framework in SMT, we easily integrate TM and EBMT into SMT via adding extra feature functions. We have shown that such integration has no harm to translation quality by comparing to an existing work. Coverage-based multiple matches can be used to further improve the system. It provides us information about how a specific source phrase is used.

As the discriminative framework naturally has the ability of dealing with a large amount of features, in the future we would like to design some EBMT-based feature functions.

Acknowledgments

We thank all the co-authors who had made their contribution to these works. They are Andy Way, Xiaofeng Wu, Santiago Cortés Vaíllo, Jun Xie and Jian Zhang.

The work in Section 4 is mainly done by Jian Zhang[5], who is the first author of the published paper. As co-authors, we thank Jian for his effort and help.

References

Axelrod, A., He, X., and Gao, J. (2011). Domain Adaptation via Pseudo In-Domain Data Selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, UK.

Biçici, E. and Yuret, D. (2014). Optimizing Instance Selection for Statistical Machine Translation with Feature Decay Algorithms. *IEEE/ACM Transactions On Audio, Speech, and Language Processing (TASLP)*.

Bisazza, A., Ruiz, N., Federico, M., and Kessler, F.-F. B. (2011). Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation. In *International Workshop on Spoken Language Translation (IWSLT)*, pages 136–143, San Francisco, CA.

Bohnet, B. (2010). Very High Accuracy and Fast Dependency Parsing is Not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.

Chang, P.-C., Galley, M., and Manning, C. D. (2008). Optimizing Chinese Word Segmentation for Machine Translation Performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232, Columbus, Ohio.

Chang, P.-C., Tseng, H., Jurafsky, D., and Manning, C. D. (2009). Discriminative Reordering with Chinese Grammatical Relations Features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, pages 51–59, Boulder, Colorado.

Chen, S. F. and Goodman, J. (1996). An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Santa Cruz, California.

Cherry, C. and Foster, G. (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*, NAACL HLT '12, pages 427–436, Montreal, Canada.

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

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

Durrani, N., Haddow, B., Heafield, K., and Koehn, P. (2013). Edinburgh's Machine Translation Systems for European Language Pairs. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria.

Eppler, E. M. D. (2013). Dependency Distance and Bilingual Language Use: Evidence from German/English and Chinese/English Data. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 78–87, Prague.

Federico, M., Bertoldi, N., and Cettolo, M. (2008). IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Proceedings of Interspeech*, pages 1618–1621, Brisbane, Australia.

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

[5] `http://www.computing.dcu.ie/~zhangj/`

Gao, Q. and Vogel, S. (2008). Parallel Implementations of Word Alignment Tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57, Stroudsburg, PA, USA. Association for Computational Linguistics.

Haddow, B. and Koehn, P. (2012). Analysing the Effect of Out-of-Domain Data on SMT systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432, Montréal, Canada.

Huang, L., Knight, K., and Joshi, A. (2006). A Syntax-directed Translator with Extended Domain of Locality. In *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 1–8, New York City, New York.

Koehn, P. (2004). Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain.

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., and Herbst, E. (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*, ACL '07, pages 177–180, Prague, Czech Republic.

Koehn, P. and Knight, K. (2003). Empirical Methods for Compound Splitting. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 187–193, Budapest, Hungary.

Koehn, P. and Senellart, J. (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.

Menezes, A. and Quirk, C. (2005). Dependency Treelet Translation: The Convergence of Statistical and Example-Based Machine-translation? In *Proceedings of the Workshop on Example-based Machine Translation at MT Summit X*.

Nakov, P. (2008). Improving English-Spanish Statistical Machine Translation: Experiments in Domain Adaptation, Sentence Paraphrasing, Tokenization, and Recasing. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 147–150, Columbus, Ohio.

Nivre, J. and Nilsson, J. (2005). Pseudo-Projective Dependency Parsing. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 99–106, Ann Arbor, Michigan.

Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Och, F. J. and Ney, H. (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, PA, USA.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (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.

Quirk, C., Menezes, A., and Cherry, C. (2005). Dependency Treelet Translation: Syntactically Informed Phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 271–279, Ann Arbor, Michigan.

Shen, L., Xu, J., and Weischedel, R. (2010). String-to-Dependency Statistical Machine Translation. *Computational Linguistics*, 36(4):649–671.

Shuyo, N. (2010). Language detection library for java.

Stolcke, A. (2002). SRILM – An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference Spoken Language Processing*, pages 901–904, Denver, CO.

Wang, K., Zong, C., and Su, K.-Y. (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.

Xie, J., Mi, H., and 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*, pages 216–226, Edinburgh, United Kingdom.

Xiong, D., Liu, Q., and Lin, S. (2007). A Dependency Treelet String Correspondence Model for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 40–47, Prague.

Appendices

A Publications

- Liangyou Li, Andy Way, Qun Liu. *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. 2014.
- Jian Zhang, Liangyou Li, Andy Way, Qun Liu. em A Probabilistic Feature-Based Fill-up for SMT. In Proceedings of the 11th Conference of the Association for Machine Translation in the Americas, Vol. 1: MT Researchers Track. Pages 96-109. Vancouver, BC, Canada. 2014.
- Liangyou Li, Jun Xie, Andy Way, Qun Liu. *Transformation and Decomposition for Efficiently Implementing and Improving Dependency-to-String Model In Moses*. In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. Pages 122-131. Doha, Qatar. 2014.
- Liangyou Li, Xiaofeng Wu, Santiago Cortés Vaíllo, Jun Xie, Andy Way, Qun Liu. *The DCU-ICTCAS MT system at WMT 2014 on German-English Translation Task*. In Proceedings of the Ninth Workshop on Statistical Machine Translation. Pages 136-141. Baltimore, Maryland, USA. 2014.