A Probabilistic Feature-Based Fill-up for SMT

Jian Zhang zhangj@computing.dcu.ie
Liangyou Li liangyouli@computing.dcu.ie
Andy Way away@computing.dcu.ie
Qun Liu qliu@computing.dcu.ie

The CNGL Centre for Global Intelligent Content,School of Computing,Dublin City University, Ireland

Abstract

In this paper, we describe an effective translation model combination approach based on the estimation of a probabilistic Support Vector Machine (SVM). We collect domain knowledge from both in-domain and general-domain corpora inspired by a commonly used data selection algorithm, which we then use as features for the SVM training. Drawing on previous work on binary-featured phrase table fill-up (Nakov, 2008; Bisazza et al., 2011), we substitute the binary feature in the original work with our probabilistic domain-likeness feature. Later, we design two experiments to evaluate the proposed probabilistic feature-based approach on the French-to-English language pair using data provided at WMT07, WMT13 and IWLST11 translation tasks. Our experiments demonstrate that translation performance can gain significant improvements of up to +0.36 and +0.82 BLEU scores by using our probabilistic feature-based translation model fill-up approach compared with the binary featured fill-up approach in both experiments.

1 Introduction

Like many machine-learning problems, Statistical Machine Translation (SMT) is a data-dependent learning approach. The prerequisite is large amounts of training data in order to generate statistical models. In general, the training data has to be sentence-aligned and bilingual. Some heuristic approaches are often used when deconstructing the training data into phrase-level representations, and the statistical models are computed based on the phrase probability distributions. The generated models are then combined in a log-linear model (Och and Ney, 2002). A basic SMT system may consist of a translation model and a language model, where the translation model provides a target-language translation *e* for a source-language sentence *f*, and the language model ensures the fluency of the target-language translation *e*.

One 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 often defined as "domain adaptation" in previous work. The underlying reasons that caused domain adaptation challenge are many, but the obvious one is that SMT system training is a complicated data-dependent processing pipeline. It often involves many efforts from various steps, for example, the phrase pair extraction step needs to be consistent with the preceding word alignment step, with one assumption made being that the context of the extracted phrases is irrelevant. In addition, the training sentences are only implicitly visible and become unnecessary once the phrase table is built. In this paper, we try to address the problem

of phrase-table extraction in a phrase-based SMT training environment, and propose a probabilistic feature-based translation model fill-up approach by creating an inheritance relationship between the extracted phrase pairs and the corresponding bilingual sentence pairs.

Domain adaptation for SMT is a well studied research field. Recently, many new ideas have been introduced, mainly regarding the data adaptation and model adaptation. Most work on data adaptation for SMT focuses on making efficient use of the training data. Lü et al. (2007) use information-retrieval techniques on a transductive-learning framework to increase the count of important in-domain training instances, which results in phrase-pair weights being favourable to the development set. Biçici and Yuret (2011) employ a feature decay algorithm which can be used in both active learning and transductive learning settings. The decay algorithm is used to increase the variety of the training set by devaluing features that have already been seen from a training set. In recent studies, a cross-entropy difference method has seen increasing interest for the problem of SMT data selection (Moore and Lewis, 2010; Axelrod et al., 2011). The training dataset is ranked using cross-entropy difference from some language models trained on in-domain or general-domain sentences. Then a threshold is set to select the *pseudo* in-domain sentences. The intuition is to find sentences as close to the target domain and as far from the average of the general-domain as possible. Later, Mansour et al. (2011) argue that "An LM does not capture the connections between the source and target words, and scores the sentences independently", and linearly interpolate IBM model 1 (Brown et al., 1993) into the cross-entropy difference framework. The translation performance is improved on both Arabic-to-English and English-to-French translation tasks compared with the standalone cross-entropy difference approach.

Applying adaptation techniques to the statistical models, especially to the translation model, is another popular approach used in domain adaptation for SMT. Some research follows the path of adding in new features into the phrase table. Chen et al. (2013) add vector similarity into the phrase table and use it as a tuning- and decoding-time feature. The similarity is computed by comparing the vectorized representation of phrase pairs extracted from the development set and the training set. Eidelman et al. (2012) achieve translation performance improvement by including a lexical weight topic feature into the translation model. The topic model used in their work is built based on the source side of the training sentences. There is also work which focuses on translation model combination. Foster and Kuhn (2007) and Koehn and Schroeder (2007) combine the translation models in a log-linear model at tuning and decoding time. Sennrich (2012) proposes an approach to interpolate the translation models based on perplexity minimization. Haddow and Koehn (2012) focus on the extracting and scoring steps when building a phrase table for SMT. One of the conclusions is that while out-of-domain data can improve the translation coverage for rare words, it may be harmful for common in-domain words. This suggests that the translations which contain a lot of in-domain evidence should be kept.

2 Related Work

The translation model fill-up approach was introduced into SMT by Nakov (2008). In his work, the phrase tables are merged by keeping all the phrase pairs unchanged from the in-domain phrase table, and only adding in the phrase pairs from the general-domain phrase tables that are not contained at the in-domain phrase table, as in (1):

$$Fill-up\{PT\} = \{PT\_in\} \cup \{PT\_out - PT\_in\} \quad (1)$$

where PT_in and PT_out are the in-domain and general-domain phrase table, respectively, and $\{PT\_out - PT\_in\}$ is the *relative complement* of PT_out in PT_in, with the original SMT translation model features from each merging phrase tables preserved. Furthermore, a new feature

value (1 or 0.5) is allocated to each phrase pair in the merged phrase table to indicate its provenance.

Bisazza et al. (2011) modify the feature value of Nakov (2008) by interpreting it differently. A scaling factor, such as 1 $(= exp(0))$ and 2.718 $(= exp(1))$, is used to define the provenance of each phrase pair in the phrase table. The fill-up model (Bisazza et al., 2011) T_F is defined as in (2):

$$\forall(\tilde{f},\tilde{e}) \in T_1 \cup T_2 : \\ \phi(\tilde{f},\tilde{e}) = \begin{cases} (\phi_1(\tilde{f},\tilde{e}), exp(0)) & if(\tilde{f},\tilde{e}) \in T_1 \\ (\phi_2(\tilde{f},\tilde{e}), exp(1)) & otherwise \end{cases} \quad (2)$$

Bisazza et al. (2011) also extend the fill-up approach into the SMT reordering model and provide a study of pruning options. The experiments show that the fill-up approach is not only able to produce comparable translation performance with log-linear combinations of translation models, but is also an approach which increases the efficiency of minimum error rate training.

3 Probabilistic Feature-based Fill-up

In this paper, we follow the previous studies (Nakov, 2008; Bisazza et al., 2011), and propose a probabilistic feature-based translation model fill-up approach for SMT. The assumption we make for our approach is that the domain information of a training sentence pair is inheritable by the extracted phrase pairs, and such an assumption is often valid in the traditional data selection research for SMT training. Data selection is often applied when in-domain training data is small and expensive to collect, but where a large amount of general-domain training data is nonetheless available. However, Haddow and Koehn (2012) point out that it might be heavy-handed if a 1-0 cutoff is used for SMT data selection, as the general-domain data can still have a contribution to the translation system. We believe that a probabilistic feature-based fill-up approach can be factored in as a soft-handed data-selection approach. Like Bisazza et al. (2011), we extend the original fill-up algorithm (Nakov, 2008), but instead of assigning firmness provenance feature values to the phrase table, we train a machine-learning algorithm to give a probability measurement with respect to the domain information to each training sentence pair. Then we use the assumption that the domain information of a training sentence pair is inheritable by the extracted phrase pairs to make such a domain-likeness feature applicable to the phrase table. The probability scale ensures the domain-likeness feature is elastic, but also under control at tuning and decoding time.

One concern is that a phrase pair in a translation table can be extracted from a number of different training 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.
- $Geometric\ Mean$: use the geometric mean value to describe the central tendency of all domain-likeness estimations.

In the rest of this paper, we describe the machine-learning algorithm used to assign the domain-likeness value in the merged phrase table, and then we introduce the feature set used to train the said learning algorithm in Section 4. Then we describe our experiments to evaluate our probabilistic feature-based translation model fill-up approach and make comparisons with the previous fill-up studies using the basic settings[1] in Section 5. Later in the paper, we make comparisons between the proposed approach with previous work on data selection (Axelrod et al., 2011) in Section 6, and provide our observations regarding the probabilistic domain-likeness feature distribution on the merged phrase table in Section 7. Finally, we give our conclusion together with avenue for future work in Section 8.

4 Support Vector Machines

4.1 SVM Algorithm

SVM is a well-known machine-learning algorithm often applied to classification or regression tasks. In classification, SVM maps a testing instance into a hyperplane which optimally separates the training data, and then outputs the predicted class label of the testing instance belongs to, the *(soft margin)* objective function is defined as (Cortes and Vapnik, 1995; Chang and Lin, 2011) in (3):

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ \text{s.t.} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, y \in \{1, -1\} \end{aligned} \tag{3}$$

where w is the weight vector, C is a tunable trade-off parameter indicating a punishment for misclassified decisions, l is the number of training instances, ξ_i is known as the slack variable, and ϕ is the kernel function mapping training instances into a high-dimensional space.

The underlying reason for using a kernel function in SVM is that the training instances in some situations are linearly non-separable and we need to improve the separability by projecting them into a high-dimensional space. In our experiments we use the Radial Basis Function (RBF) kernel for SVM training and predicting, defined as in (4):

$$exp\{-\gamma|u - v|^2\} \tag{4}$$

The gamma parameter γ is a tunable variable which adjusts the width of RBF.

As SVM predicts class labels only, Chang and Lin (2011) extend the approach proposed by Wu et al. (2004) to give a probability estimation for every prediction. In our work, we use the predicted probability to indicate the domain-likeness estimation.

4.2 SVM Feature Set

It is worth recalling that our probabilistic feature-based fill-up approach is based on the assumption that the domain information of a training sentence pair can be inherited by the extracted phrase pairs, and such an assumption is often applied in SMT data selection algorithms for domain adaptation. In our case, if we are able to assign a probabilistic domain-likeness value to each training sentence, then to include them as a new decoding feature into the fill-up phrase table is effortless. Thus, we can transfer our objective into assigning the domain-likeness estimation to the SMT training sentences.

The cross-entropy, which is defined as in (5):

$$H(p_{LM}) = -\frac{1}{n} \sum_{i=1}^{n} log p_{LM}(w_i|w_1, ..., w_{i-1}) \tag{5}$$

[1]The fill-up (Bisazza et al., 2011) provides several pruning options. There is also a cascaded fill-up method applicable for more than one general-domain phrase model. We do not make comparisons for these cases.

has been used as a strong domain indicator in much adaptation research (Klakow, 2000; Gao et al., 2002; Moore and Lewis, 2010; Axelrod et al., 2011). In equation (3), n is the number of words w in a sentence. However, in our work, we use the transformation of cross-entropy, known as perplexity, which is defined as in (6),

$$Perplexity = 2^{H(p_{LM})} \tag{6}$$

We take inspiration from the previous works in Axelrod et al. (2011), and design three sets of SVM training 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 perplexity difference and the target-side perplexity difference in this feature set.

5 Experiment

5.1 Corpora

The experiments in this paper use data from WMT07, WMT13 and IWLST11 translation tasks. We choose our experiments on the French-to-English language pair. We first perform some standard data cleaning steps, including tokenization, punctuation normalization, replacement of special characters, lower casing and long sentence removal ($<$0 or $>$80), resulting in the preprocessed data summarized in Table 1. We use scripts provided within Moses 1.0 translation system framework (Koehn et al., 2007)[2] for all cleaning steps.

Corpus	**Train**	**Tune**	**Test**
News Commentary (*nc_2007*)	42,884	1,064 (nc-devtest200)	2,007 (news-test2007)
Europarl (*ep_2007*)	1,257,436	n/a	n/a
TED (*ted_11*)	106,642	934 (dev2010)	1,664 (tst2010)
news-commentary-v9 (*nc_v9*)	181,274	n/a	n/a

Table 1: SMT training corpus statistics

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(nc_2007,ep_2007)* and *prob-fill-up_heuristic(ted_11,nc_v9)*, where *heuristic* refers to the heuristics stated in Section 3 of this paper. The experimental design is to assess our approach in both of the following situations: (i) general-domain dataset being significant larger than the in-domain data, and (ii) the two datasets being similar in size, as seen in Table 1.

[2] `http://www.statmt.org/moses/`

5.2 SVM training

We uses the R language package *e1071* (Dimitriadou et al., 2009)[3] to train the SVM algorithm, where the *e1071* in R language is an interface to the libsvm (version 2.6) (Chang and Lin, 2011) implementation. 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.

A set of high-quality training data for tasks like classification is a luxury in machine-learning, and such datasets often cannot be obtained automatically. 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. One solution is to rank the general-domain instances with respect to the known in-domain information, and then mark the most distant partition instances as the opposite of the in-domain class for SVM training. Such a solution can create a clear boundary in the SMT training set, but there is a danger of causing the SVM training data to be of low variance and high bias. The reason for this is that a similar amount of SVM training instances from both labeled classes are suggested to be used in order to set up a fair training condition. However, in the domain-adaptation context, where only a small amount of in-domain instances and a large amount of general-domain instances are available, we are restricted to selecting only a limited number of SVM training instances. The size limitation and the ranking selection used may lead the SVM training instances to be of low variance and high bias. In addition, we also have the prior knowledge of the predicting instances available before the SVM is trained, but it is unfortunate that such knowledge is ignored. In fact, the SVM in our case prefers to be trained on the two classes of instances that represents the average of the general-domain dataset and the in-domain dataset. Then the probability prediction produced by such an SVM can indicate the distance of a predicting instance from those two classes. Thus, we simply randomly select *M* number of general-domain and in-domain sentences as SVM training instances in our experiments.

To extract features for the selected SVM training data, we randomly select an equal number (size *N*) of sentences from the in- and the general-domain dataset and train an *n*-gram language model, where $n = \{2 \ldots 5\}$, then extract the perplexity features for each *n* setting. The language model training at this step uses the same restrictions as in Moore and Lewis (2010), where a token is treated as an instance of $<UNK>$ unless it appears at least twice at the in-domain training dataset. We keep *T* number of SVM training sentences to tune the parameters in equations (3) and (4). We test the accuracy of the trained SVM using the corresponding SMT development data. The data used for SVM training, language model training and SVM tuning are summarized in Table 2. The SVM-tuned parameters are presented in Table 3. We use the open source IRSTLM toolkit (Federico et al., 2008) for language model training and KenLM (Heafield, 2011) to compute the sentence perplexity.

Experiment	***M***	***N***	***T***
prob-fill-up(nc_2007,ep_2007)	42,884	40,000	2,884
prob-fill-up(ted_11,nc_v9)	50,000	45,000	5,000

Table 2: SVM data statistics, where *M*,*N* and *T* are the data sizes (in sentences) used for training, tuning and testing, respectively.

[3] `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

Experiment	*C*	γ	Accuracy
prob-fill-up(nc_2007,ep_2007)	16	0.125	0.8139
prob-fill-up(ted_11,nc_v9)	2	0.03125	0.8565

Table 3: SVM-tuned parameters values *C* and γ, where *C* is the trade-off parameter in equation (3), and γ adjusts the width of RBF in equation (4).

5.3 Translation System Training

All SMT systems in our experiments are trained using the phrase-based SMT with Moses 1.0 framework. The reordering model is not included in our translation system since we are interested only in measuring the system effects coming from translation models. We use the word aligner MGIZA++ (Gao and Vogel, 2008) for word alignment in both translation directions, and then symmetrize the word alignment models using the heuristic of grow-diag-final-and. We use all five default Moses 1.0 translation model features. The translation systems are tuned with minimum error rate training (Och, 2003) using case-insensitive BLEU (Papineni et al., 2002) as the optimization measure. A 5-gram language model is trained with the open source IRSTLM toolkit using all the available target sentences in each of the fill-up experiment scenarios. We use the Moses default language model toolkit KenLM at the tuning and decoding time.

5.4 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 1.0 framework. Tables 4 and 5 report our results using case-insensitive BLEU on the corresponding test sets. We use † to indicate where the probabilistic feature-based fill-up approach systems (*prob-fill-up_heuristic(experiment)*) achieve significant improvement (Koehn, 2004) compared with the baseline systems at the level $p = 0.01$ level with 1000 iterations.

System	Test (news-test2007)
fill-up(nc_2007,ep_2007)	28.01
prob-fill-up_Min(nc_2007,ep_2007)	28.03
prob-fill-up-Arithmetic_Mean(nc_2007,ep_2007)	28.21
prob-fill-up-Geometric_Mean(nc_2007,ep_2007)	28.37†

Table 4: *prob-fill-up_heuristic(nc_2007,ep_2007)* 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(nc_2007,ep_2007)* experiment in Table 4 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 seen in Table 5, 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

System	Test (tst2011)
fill-up(ted_11,nc_v9)	30.82
prob-fill-up_Min(ted_11,nc_v9)	30.73
prob-fill-up_Arithmetic_Mean(ted_11,nc_v9)	31.64†
prob-fill-up_Geometric_Mean(ted_11,nc_v9)	31.21†

Table 5: *prob-fill-up_heuristic(ted_11,nc_v9)* experiment BLEU scores on testing data, the significance testing at the level $p = 0.01$ level with 1000 iterations.

qualify as statistically significant improvements over the baseline system at $p = 0.01$ level. The *prob-fill-up_Min(ted_11,nc_v9)* system underperforms the baseline system by about 0.1 absolute BLEU score difference.

Overall, our approach is able to significantly improve upon the baseline translation performance in both of the designed testing scenarios.

6 Data selection

In this section, we compare our probabilistic feature-based fill-up approach with the data selection approach proposed in Axelrod et al. (2011). In general, data selection is one of the standard approaches used in SMT training when out-of-domain or general-domain data is available. It is often required to train many SMT systems in order to find the most appropriate proportion of general-domain data to include and obtain the best performance from it. In this experiment, 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 (7), with the ranking in reverse order:

$$PPL - DIFF = \frac{[PPL_{I\_src(S)} - PPL_{O\_src(S)}]}{length(S)} + \frac{[PPL_{I\_tgt(T)} - PPL_{O\_tgt(T)}]}{length(T)} \quad (7)$$

where *S* and *T* are the source and target sentences, respectively. The language models described in Section 5.2 are used to compute perplexities. The top p proportion of the ranked general-domain corpus is selected, and concatenated with the in-domain corpus. The concatenation is then used to train the data selection systems. We employ the same experimental settings described in Section 5.3 for this experiment, with the word alignments computed in advance using the combination of all in- and general-domain data. The tuning and test datasets described in Table 1 are also taken in order to compare with the experiment results described in Section 5.4.

Figures 1 and 2 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 *ep_2007* and 65% of *nc_v9* 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 of the general-domain data, achieving significantly better translation results (Table 4) on the *(nc_2007,ep_2007)* dataset and comparable translation results (Table 5) on the *(ted_11,nc_v9)* dataset.

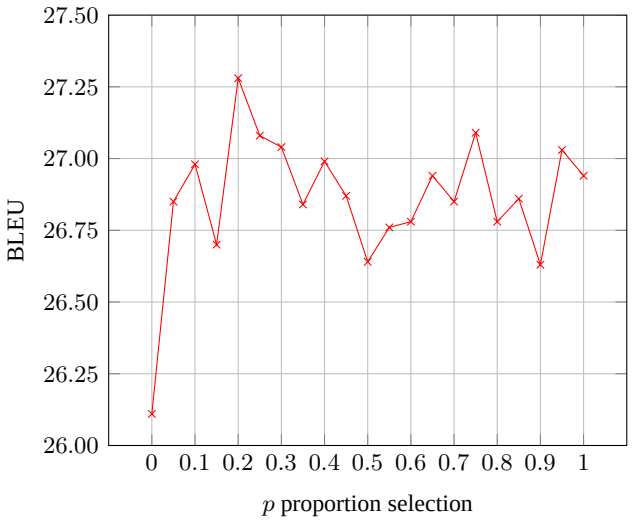

Figure 1: BLEU scores with different p proportion of data selection on *(nc_2007,ep_2007)* dataset.

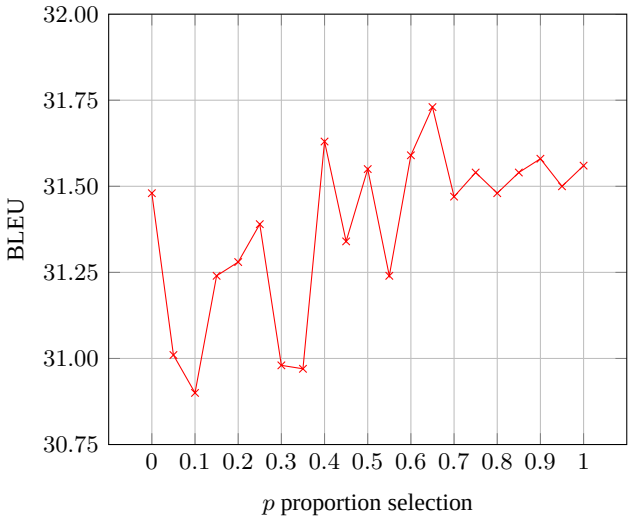

Figure 2: BLEU scores with different p proportion of data selection on *(ted_11,nc_v9)* dataset.

7 Domain-likeness Distribution

In this section, we study the distribution of the domain-likeness feature added into the final merged phrase table. The main difference between our approach with the previous fill-up methods is the interpretation of the additional features employed. A learned probabilistic domain-likeness feature is used by our approach, while a binary provenance indicator is applied in previous work. It is easy to establish that the in-domain part of the produced phrase tables is identical in our and previous work, and that the total number of phrase entries is also the same. Thus, we mainly focus on the general-domain phrase entries in this section. We take the *prob-fill-up_heuristic(ted_11,nc_v9)* experiment in the previous section as the case study.

The *prob-fill-up_heuristic(ted_11,nc_v9)* experiment merges *pt(ted_11)* and *pt(nc_v9)* phrase tables. 5,790,068 in-domain phrase entries from *pt(ted_11)* are kept, and 12,915,649 general-domain phrase entries from *pt(nc_v9)* are used to fill-up. 236,779 of the phrase entries from *pt(nc_v9)* conflict with the phrase entries in *pt(ted_11)*, and are neglected in the final produced phrase table. The final merged phrase table contains 18,468,938 phrase entries in the *prob-fill-up_heuristic(ted_11,nc_v9)* experiment, where the standalone phrase table using the concatenated ted_11 and nc_v9 corpus produces 18,339,548 phrase pairs.

Interval Group	**# of phrases** Min	**# of phrases** $Arithmetic\_Mean$	**# of phrases** $Geometric\_Mean$
0.95 ~ 1.00	1,301,571	1,301,803	1,301,820
0.90 ~ 0.95	29,085	29,197	29,209
0.85 ~ 0.90	20,117	20,229	20,254
0.80 ~ 0.85	16,272	16,335	16,366
0.75 ~ 0.80	15,565	15,625	15,675
0.70 ~ 0.75	14,041	14,164	14,352
0.65 ~ 0.70	12,816	12,966	13,747
0.60 ~ 0.65	12,635	12,889	13,595
0.55 ~ 0.60	12,536	12,938	13,759
0.50 ~ 0.55	11,562	13,121	21,299
0.45 ~ 0.50	14,673	15,930	33,106
0.40 ~ 0.45	13,596	15,539	20,530
0.35 ~ 0.40	16,060	43,168	22,923
0.30 ~ 0.35	17,022	26,438	34,956
0.25 ~ 0.30	20,564	29,720	34,802
0.20 ~ 0.25	24,397	47,674	43,848
0.15 ~ 0.20	31,233	56,000	55,217
0.10 ~ 0.15	45,590	81,080	79,150
0.05 ~ 0.10	88,412	146,956	140,063
0.00 ~ 0.05	5,916,294	5,722,269	5,709,370

Table 6: Filtered *prob-fill-up_heuristic(ted_11,nc_v9)* phrase table entry counts with intervals of 0.05 according to SVM-assigned domain-likeness feature value.

To demonstrate the distribution of the phrase pairs in the merged phrase table, we first group the phrase entries in the merged phrase tables (filtered using the corresponding test set) with intervals of 0.05 according to the domain-likeness feature value. We can observe in Table 6 that the SVM predictions fall mostly into the 0.00 ~ 0.05 or 0.95 ~ 1 intervals. We think that the prediction follows the natural composition of the general-domain dataset, so the composition can be described as consisting of some of the target unrelated sentences, some of the

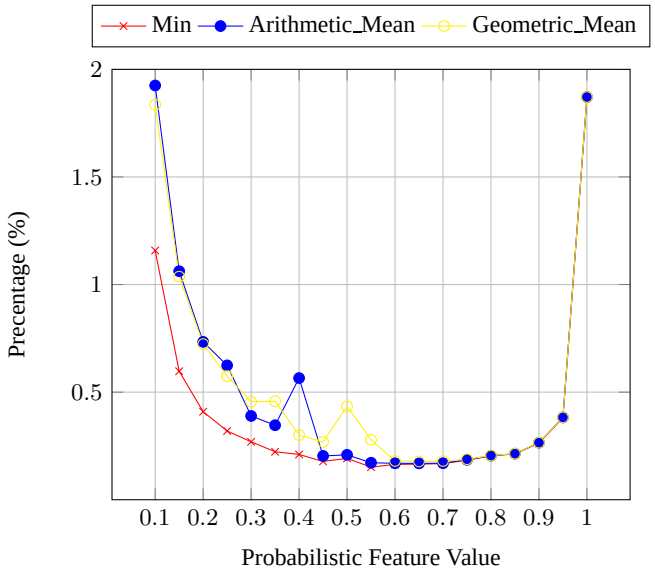

Figure 3: The distribution of Min,$Arithmetic\_Mean$ and $Geometric\_Mean$ phrase pairs contribution comparison: X-axis represents the range from 0.10 to 1.00. Y-axis represents the percentage of phrase entries to the overall testing data filtered phrase table.

mixed domain sentences and some of the in-domain sentences. The range between 0.05 ~ 0.95 also draws our attention. All three heuristic functions create similar numbers of phrase entries for each interval group at the upper bound range: 0.70 ~ 1.00. This may be evidence that there is only 0.92 BLEU score difference between the best- and worst-performed probabilistic feature-based fill-up systems in Table 5 since the upper bound range is the closest to the target translation domain. Later, the $Geometric\_Mean$ system acts more aggressively and there is a dramatic increase in the quality of phrase pairs at the intervals of 0.45 ~ 0.50. We think that this interval is the most uncertain region in the general-domain dataset given the knowledge inferred by the corresponding heuristic functions. A similar increase also can be found in the $Arithmetic\_Mean$ system at the intervals of 0.35 ~ 0.40, but the increasing curve is sharper compared with the growth in $Geometric\_Mean$. The lower bound range in Table 6 is in a very mixed situation.

The graph in Figure 3 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 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 also that the 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.

8 Conclusion and Future Work

In this paper, we addressed the inaccurate assumption introduced at the phrase extraction step for phrase-based SMT training. We extended the fill-up phrase-table merging approach by assigning a domain-likeness probabilistic feature. We described the rationale behind our probabilistic feature-based fill-up approach and explained our intuitions regarding the SVM feature set. We also designed two experimental scenarios, showing 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 paper is still preliminary and can be extended further. We have not carried out experiments regarding any implication between the SVM performance and the SMT translation performance; our SVM features are purely inspired by the previous data selection studies and can also be more elegant. In future work, we would like to carry out such studies. We would also like to experiment on a reordering model fill-up and introduce more domain-oriented SVM training features. The proposed probabilistic feature-based fill-up approach can also be viewed as a domain adaptation approach, where bilingual in-domain training sentences are unavailable, but where a large amount of general-domain bilingual training sentences is easy to obtain. We can train the SVM algorithm to assign the domain-likeness feature using the source and the target monolingual in- and general-domain data to the general-domain only phrase table. Thus the general-domain-only phrase table can gain some domain knowledge at decoding time.

9 Acknowledgements

This research is supported by the Science Foundation Ireland (Grant 12/CE/I2267) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University.

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. (2011). Instance Selection for Machine Translation using Feature Decay Algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283, Edinburgh, UK.

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.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chen, B., Kuhn, R., and Foster, G. (2013). Vector Space Model for Adaptation in Statistical Machine Translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1285–1293, Sofia, Bulgaria.

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

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., Weingessel, A., and Leisch, M. F. (2009). Package 'e1071'. *R Software package, available at http://cran.rproject.org/web/packages/e1071/index.html*.

Eidelman, V., Boyd-Graber, J., and Resnik, P. (2012). Topic Models for Dynamic Translation Model Adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 115–119, Jeju, Korea.

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.

Foster, G. and Kuhn, R. (2007). Mixture-Model Adaptation for SMT. In *Proceedings of the Second Association for Computational Linguistics Workshop on Statistical Machine Translation*, Prague, Czech Republic.

Gao, J., Goodman, J., Li, M., and Lee, K.-F. (2002). Toward a Unified Approach to Statistical Language Modeling for Chinese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 1(1):3–33.

Gao, Q. and Vogel, S. (2008). Parallel Implementations of Word Alignment Tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57.

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.

Heafield, K. (2011). KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland.

Klakow, D. (2000). SELECTING ARTICLES FROM THE LANGUAGE MODEL TRAINING CORPUS. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 1695–1698.

Koehn, P. (2004). Statistical Significance Tests for Machine Translation Evaluation. In *The 2004 Conference on Empirical Methods on Natural Language Processing*, 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., et al. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, pages 177–180, Prague, Czech Republic.

Koehn, P. and Schroeder, J. (2007). Experiments in Domain Adaptation for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227, Prague, Czech Republic.

Lü, Y., Huang, J., and Liu, Q. (2007). Improving Statistical Machine Translation Performance by Training Data Selection and Optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 343–350, Prague, Czech Republic.

Mansour, S., Wuebker, J., and Ney, H. (2011). Combining Translation and Language Model Scoring for Domain-Specific Data Filtering. In *Proceedings of the Eighth International Workshop on Spoken Language Translation (IWSLT)*, pages 222–229, San Francisco, CA, USA.

Moore, R. C. and Lewis, W. (2010). Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden.

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.

Och, F. J. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

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.

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, PA, USA.

Sennrich, R. (2012). Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549, Avignon, France.

Wu, T.-F., Lin, C.-J., and Weng, R. C. (2004). Probability Estimates for Multi-class Classification by Pairwise Coupling. *Journal of Machine Learning Research*, 5(4):975–1005.