Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation

Arianna Bisazza, Nick Ruiz, Marcello Federico

FBK - Fondazione Bruno Kessler
Via Sommarive 18, 38123 Povo (TN), Italy
{bisazza,nicruiz,federico}@fbk.eu

Abstract

This paper compares techniques to combine diverse parallel corpora for domain-specific phrase-based SMT system training. We address a common scenario where little in-domain data is available for the task, but where large background models exist for the same language pair. In particular, we focus on phrase table *fill-up*: a method that effectively exploits background knowledge to improve model coverage, while preserving the more reliable information coming from the in-domain corpus. We present experiments on an emerging transcribed speech translation task – the TED talks. While performing similarly in terms of BLEU and NIST scores to the popular log-linear and linear interpolation techniques, filled-up translation models are more compact and easy to tune by minimum error training.

1. Introduction

Statistical machine translation (SMT) systems have the important potential of being tailored to the specific type of language used in a task. At the same time, SMT performance heavily depends on the quantity of the available training material. The strain to balance these contrasting needs has motivated a large body of work in domain adaptation. In this work, we aim at increasing the coverage of a small but precise in-domain model. To this end, we assume that all the information coming from our primary source should be preserved as is, and use the secondary sources only to 'fill the gaps'.

The idea of *fill-up* goes back to Besling and Meier [1], which addressed the problem of language model adaptation for speech recognition, and was recently introduced in SMT by Nakov [2]. The original method was conceived by [1] to train speaker-dependent dictation systems and proved to outperform classical linear interpolation. In that context, the primary source of information was, naturally, the set of sentences uttered by a given speaker, as opposed to all the others. The SMT scenario that we are addressing is of course different, but here we can use our prior knowledge of the task to make assumptions on the relevance of the available corpora. For a practical example, consider the TED[1] talks translation task [3]. The training material provided for the IWSLT11 evaluation campaign consists of a rather small corpus of TED talks (100K parallel sentences) plus a variety of large out-of-domain corpora: news stories, UN documents and European Parliament proceedings. The variety of topics covered by the talks makes this a very challenging task, for which the TED corpus alone cannot ensure sufficient coverage. We then take the TED-only phrase table as our core (primary) model and use models trained on the other corpora to augment this with new phrase pairs. The feature values of the phrase pairs found in the TED table remain untouched. Finally, an additional feature is added to each phrase pair to distinguish newly added pairs from in-domain pairs. The resulting model can be tuned as usual, with the last feature acting as a scaling factor for out-of-domain translation scores. Through a simple reliability criterion, we can thus obtain models that are less redundant and easier to tune.

The rest of this paper is organized as follows. After a review of relevant work, we describe in detail the fill-up technique and present possible refinements and extensions. In the experimental section, we apply the fill-up technique to two TED translation tasks and compare it with the two most popular methods for phrase table combination: linear and log-linear interpolation.

2. Previous Work

Previous work on domain adaptation for SMT has focused on techniques for selecting parallel or monolingual in-domain training data (e.g. [4]), as well as methods for combining models trained independently on in-domain and on out-of-domain data. Given the scope of our work, we review here only approaches for combining in-domain and out-of-domain (background) translation models.

Existing approaches combine different sources either at the data level or at phrase-table level. Adaptation at phrase-table level is either done off-line, typically by a linear mixture of weights, or at decoding-time through a log-linear combination. In the former case, a generative model and maximum likelihood estimation are employed; in the latter case weights of the log-linear interpolation are typically learned discriminatively by directly optimizing the performance of the SMT decoder.

In [5] a mixture-model approach is proposed, with weights depending on some text distances between in-

[1] http://www.ted.com/talks

domain data and the mixture components. The authors explored different choices: cross-domain and dynamic adaptation; linear and log-linear mixtures; different text distance metrics and methods to map them to linear mixture weights. For log-linear mixtures, weights were estimated globally with the other features of the phrase-based model, through minimum error rate training [6]. Notice that the employed system used a relatively small number of features: two probabilities for each phrase table, one for each language model, a length penalty, and a distortion model. Reported results show improvements by the linear and log-linear mixtures over a baseline trained on the union of all training data. Remarkably, best results with the linear mixture were obtained using uniform weights.

In [7] a phrase-based SMT system trained on Europarl data is adapted to the news domain by integrating it with language and translation models, explicitly trained on in-domain data. In particular, the in-domain phrase-table was added to the global log-linear model. As a difference with [5], phrase-pairs are here scored with four translation probabilities and four reordering probabilities, thus resulting in a significantly larger set of feature weights to be trained.

In [8] in-domain and out-of-domain phrase-tables are also combined using a two-component linear mixture. Extensive experiments are reported with different data-selection criteria and empirical weight settings. The contribution of the mixture approach is relevant and quite stable within a large interval of weight values, centered around 0.5.

Very recently [9] proposed novel data selection criteria to extract "pseudo in-domain" data from a large background parallel corpus which is then used either to train a domain-specific SMT system, or to adapt a generic SMT system via linear and log-linear mixtures, similarly to [5] but with a feature set similar to that used in [7]. In the reported experiments, the log-linear method outperformed the linear mixture adaptation method and both methods outperformed the in-domain and generic baselines.

In [10] a corpus identifier is introduced to distinguish parallel in-domain data from out-of-domain data in a factored translation model. Each target word is assigned an id tag corresponding to the part of the corpus from which it belongs. Three additional translation model features are introduced to compute the probability of corpus id tags being generated given the source phrase, as well as the source and target phrase probabilities, given the corpus id tags. The incorporation of corpus id tags promotes the preference of phrase pairs from a specific domain.

Finally, the system description paper [2] recently introduced a phrase-table merging approach practically equivalent to our fill-up technique, but with a slightly different definition of the additional feature used to indicated the origin of each phrase-pair. In [2] this feature assumes values 0.5 and 1 in the log-space, to indicate, respectively, in-domain and out-of-domain phrase-pairs. In our implementation, values of 0 and 1 respectively are assumed instead. In this way, the additional feature weight can be interpreted as a scaling factor for the out-of-domain probabilities. We provide here more background and a detailed description of the method, in addition to testing several pruning options when combining the phrase tables. Moreover, we implemented the fill-up adaptation method on a popular open source SMT platform, tested it on a speech translation task and compared it with two other popular data combination techniques.

3. Phrase table fill-up

The fill-up technique is applied after a standard phrase-based SMT training procedure, just before weight optimization. First, separate translation models are built from in-domain and background data. This implies word alignment[2], phrase extraction and phrase scoring. In standard adaptation scenarios, background data is augmented with in-domain data; however, in the fill-up case, the background table is merged with the in-domain table by adding only new phrase pairs that do not appear in the in-domain table. Formally, let T_1 and T_2 be the in-domain and the background phrase tables, respectively. The translation model assigns a feature vector to each phrase pair $\phi(\tilde{f},\tilde{e})$, where $\tilde{f}$ and $\tilde{e}$ are respectively the source and target phrases. Namely, in the model we are using [11], five features are defined for each phrase pair:

$$\phi(\tilde{f},\tilde{e}) = (P_{ph}(\tilde{e}|\tilde{f}), P_{ph}(\tilde{f}|\tilde{e}), P_{lex}(\tilde{e}|\tilde{f}), P_{lex}(\tilde{f}|\tilde{e}), pp(\tilde{f}|\tilde{e}))$$

where P_{ph} refers to the phrase translation probability, P_{lex} is the lexical weighting probability, and pp is a constant phrase penalty that serves to adjust the degree of phrase segmentation (typically $pp = \exp(1)$). Then, the filled-up model T_F is defined as follows:

$$\forall(\tilde{f},\tilde{e}) \in T_1 \cup T_2 :$$

$$\phi_F(\tilde{f},\tilde{e}) = \begin{cases} (\phi_1(\tilde{f},\tilde{e}), \exp(0)) & \text{if } (\tilde{f},\tilde{e}) \in T_1 \\ (\phi_2(\tilde{f},\tilde{e}), \exp(1)) & \text{otherwise} \end{cases}$$

The entries of the filled-up model correspond to the union of the two phrase tables, while the scores are taken from the more reliable source whenever possible. To keep track of a phrase pair's provenance, we add a binary feature[3] that fires if the phrase pair comes from the background table. It is easy to show that the weight assigned to this feature acts as a scaling factor for the out-of-domain translation scores. In fact, minimum error training will determine how the latter should be penalized.

As opposed to the log-linear combination of phrase tables, fill-up leads to a smaller feature vector, while maintain-

[2]To obtain a more accurate word alignment, this first step can be performed on the concatenation of all corpora, provided that phrase extraction and scoring are carried out separately on each corpus.

[3]We apply the exponential function to binary features to neutralize the log function that is applied to all features participating in the log-linear model.

ing a way to promote one set of phrase pairs with respect to the other.

3.1. Reordering table fill-up

When combining multiple phrase tables, one has generally to deal with phrase reordering models as well. Our system includes a popular lexicalized reordering model [12, 13, 14] whose entries are those of the phrase table trained on the same corpus, and whose features are reordering probabilities with three possible values: *monotonic* if immediately following the last translated phrase, *swap* if immediately preceding it or else *discontinuous*. The phrase table fill-up technique can be seamlessly applied to this type of reordering model, with the only difference that no additional feature is introduced.

3.2. Pruning options

We explored several pruning options to limit the new translation model size:

- *NewSourceMaxLength*: set a maximum length for the source side of new (background) phrase pairs;
- *OnlyNewSourcePhrases*: take new phrase pairs only if their source side is not covered by the in-domain model. In other words, do not add background translations of known source phrases;
- *OnlyNewSourceWords*: take new phrase pairs only if they contain a source word that does not appear in the in-domain model's vocabulary.

Empirical results for each option are discussed in the experiments section.

3.3. Fill-up cascade

If more than one out-of-domain dataset is available, and if an order of relevance/reliability can be established among them, the fill-up method can be applied in cascade. For each out-of-domain model a new binary feature is added, so that minimum error rate training can learn to weigh different data collections independently. Assuming the same number of phrase and reordering tables, if $|T|$ is the number of phrase tables to be merged (including the in-domain one), $|\phi|$ is the original size of the translation feature vector, and $|\rho|$ is the size of the reordering feature vector, then the final number of features will be:

$$|\phi| + |\rho| + (|T| - 1)$$

whereas with log-linear combination it would be:

$$(|\phi| + |\rho|) \times |T|.$$

In our setting, three phrase tables with size 5 and three reordering tables of size 6 yield only 13 weights to tune instead of 33.

4. Interpolation techniques

4.1. Linear interpolation

The simplest mixture model is a linear mixture, defined as:

$$p(x \mid h) = \sum_c \lambda_c p_c(x \mid h),$$

where $p(x \mid h)$ refers to the translation model or the reordering model and $p_c(x \mid h)$ is the component c corresponding to the translation model in the mixture. Each component c receives an associated weight, λ_c, such that $\sum_c \lambda_c = 1$. It is also common to perform linear interpolation on reordering models. The downside of linear interpolation is that there is not a consensus on the best technique to optimize the mixture weights. In our experiments we use uniform weights, as this often appears in the adaptation literature as a competitive baseline.

4.2. Log-linear interpolation

The log-linear combination of translation models is another approach to domain adaptation that is discussed in [5, 7]. Additional translation models are incorporated globally in the log-linear model by adding additional features corresponding to the translation model's phrase table and reordering models. Feature weights are optimized altogether on a development set by a standard minimum error training procedure.

When decoding with multiple phrase tables, multiple translation options and decoding paths are generated for the same phrase pair, if they appear in more than one table[4]. This behavior may interfere negatively with pruning parameters such as the maximum number of translation options and beam size.

5. Experiments

We evaluate fill-up, log-linear and linear interpolation on the TED task, in two different language pairs: Arabic-to-English and English-to-French. Training and test data were provided by the organizers of the IWSLT11 evaluation, and are summarized in Table 1[5]. The tuning (dev2010) and test (test2010) sets have one reference translation.

Concerning preprocessing we apply standard tokenization to the English and French data, while for Arabic we use our in-house tokenizer that also removes diacritics and normalizes special characters and digits. Arabic text is then segmented with AMIRA [16] according to the ATB scheme[6].

For both language pairs, we set up a standard phrase-based system using the Moses toolkit [15]. The decoder features a statistical log-linear model including one or more

[4]These observations refer to the Moses decoder [15], but we are not aware of other decoders having a different solution to this problem.

[5]Europarl corpus was also available for English-to-French, but we did not use it in our experiments.

[6]The Arabic Treebank tokenization scheme isolates conjunctions *w+* and *f+*, prepositions *l+, k+, b+*, future marker *s+*, pronominal suffixes, but not the article *Al+*.

Table 1: *IWSLT11 training and test data statistics: number of sentences $|S|$, number of tokens $|W|$ and average sentence length $\bar{\ell}$. Token numbers refer to the target language, except for the test sets.*

Corpus		$\|S\|$	$\|W\|$	$\bar{\ell}$
AR-EN	TED	90K	1.7M	18.9
	UN	7.9M	220M	27.8
EN	TED	124K	2.4M	19.5
	NEWS	30.7M	782M	25.4
AR test	dev2010	934	19K	20.0
	test2010	1664	30K	18.1
EN-FR	TED	105K	2.0M	19.5
	UN	11M	291M	26.5
	NEWS	111K	3.1M	27.6
FR	TED	107K	2.2M	20.6
	NEWS	11.6M	291M	25.2
EN test	dev2010	934	20K	21.5
	test2010	1664	32K	19.1

Table 2: *Phrase table statistics (in millions of phrase pairs) of the Arabic-English and English-French training corpora. The common phrases and new translations are reported with respect to the TED phrase table.*

Phrase set	Millions of ph. pairs	
	Ar-En	En-Fr
$\|T_{ted}\|$	2.8	2.6
$\|T_{un}\|$	132.9	130.0
$\|T_{ted} \cap T_{un}\|$	0.1	0.6
$\|$*NewSourceMaxLength=4* $(T_{un})\|$	50.1	50.0
$\|$*OnlyNewSourcePhrases*$(T_{un})\|$	131.6	128.3
$\|$*OnlyNewSourceWords*$(T_{un})\|$	32.1	0.7
$\|T_{news}\|$	–	2.7
$\|T_{ted} \cap T_{news}\|$	–	0.1
$\|$*NewSourceMaxLength=4* $(T_{news})\|$	–	1.3
$\|$*OnlyNewSourcePhrases*$(T_{news})\|$	–	2.5
$\|$*OnlyNewSourceWords*$(T_{news})\|$	–	0.02

phrase translation models, target language models, a phrase reordering model [12, 13], distortion, word and phrase penalties.

In the Arabic-English task, we use a hierarchical reordering model [14], while in the English-French task we use a default word-based bidirectional extraction model. For each target language, two 5-gram language models are trained independently on the monolingual TED and NEWS datasets, and log-linearly combined at decoding time. The distortion limit is set to the default value of 6. As proposed by [17], statistically improbable phrase pairs are removed by all our phrase tables (before merging). The Arabic-English systems use cased translation models, while the English-French systems use lowercased models and a standard recasing post-process.

Word alignments are computed by GIZA++ [18] on the concatenation of all data. Consequently, phrase extraction and scoring are carried out separately on each corpus. Table 2 provides summary statistics on the phrase overlaps of the NEWS and UN phrase tables with respect to the TED phrase table.

Note that, in this work, we do not evaluate the contribution of the reodering model in isolation. Thus, in each experiment, the same data combination technique is used to build both translation and reordering models.

As suggested by [19], we use approximate randomization to test whether differences among system performances are statistically significant[7].

[7]Significance tests were computed with the *Multeval* toolkit: https://github.com/jhclark/multeval

5.1. Arabic to English

We apply fill-up and plug the resulting phrase and reordering tables (5+1 and 6 features respectively) to the decoder. The global feature vector for each experimental setting is then optimized by minimum error rate training (MERT) [6]. Table 3 presents translation quality results in terms of BLEU and NIST scores, using different data combination techniques: *concat* stands for a unique translation (and reordering) model estimated on the concatenation of all data, *linear* is the linear interpolation of TED and UN models with uniform weights and *logli* is a decoding-time log-linear combination. The rows named *fillup* show results obtained with different fill-up pruning options.

We can see that the addition of background data to the TED in concatenation mode sensibly degrades the performance from 24.96 to 23.45 BLEU (statistically significant with $p < .01$). This is due to the fact that the background data overwhelms the in-domain data and it is not possible to scale the probabilities of one corpus with respect to the other.

All other combination techniques, instead, yield improvements with respect to the TED-only model, with log-linear combination and fill-up emerging as the best systems at the $p < .01$ level. The gain achieved by fill-up over log-linear is, however, not statistically significant.

Concerning the proposed fill-up pruning options, we note that the best BLEU result (25.88) is obtained with the pruning of long background phrases (more than 4 source words), while the highest NIST (6.515) is obtained with un-pruned fill-up. However, these differences are not significant. We prefer the source-length pruned model because of its more manageable size: 54.1M entries instead of 135.6M. As for the other pruning options, they both have a negative impact on translation quality: *onlyNewSrcPhrases* yields a BLEU score of 25.72, suggesting that new translation of known

Table 3: *%BLEU|NIST scores on Arabic-English TED, using different data combination techniques.*

Translation model	fill-up pruning	test2010
only TED	—	24.96 | 6.434
concat(TED+UN)	—	23.45 | 6.130
logli(TED+UN)*	—	25.62 | 6.474
linear(TED+UN)	—	25.15 | 6.401
fillup(TED+UN)	none	25.78 | 6.515
fillup(TED+UN)	newSrcMaxLength=4	25.88 | 6.512
fillup(TED+UN)	onlyNewSrcPhrases	25.72 | 6.505
fillup(TED+UN)	onlyNewSrcWords	25.37 | 6.446
fillup-n(TED+UN)	newSrcMaxLength=4	25.46 | 6.451

*MERT didn't converge by the 25th iteration.

source phrases also help to improve the model and thus shouldn't be pruned. The last option tested, *onlyNewSrcWords*, yields the worst fill-up result (25.37), probably due to the harshness of this pruning criterion that mantains only one fourth of the new phrase pairs found in UN (see Table 2).

Additionally, we compare our phrase penalties of 0 and 1 with the penalties of 0.5 and 1, originally described in [2]. The latter configuration (*fillup-n*) obtains a lower score (25.46 as opposed to 25.88, $p < .05$).

As previously stated, fill-up is not significantly better than log-linear; however, the drawback of the latter method is apparent in the behavior of MERT. As can be seen in Figure 1, MERT converges much faster with the filled-up models, probably due to the lower number of features. Indeed, MERT is known to be best suited to tune a limited number of weights (see for instance [20]). In the setting of two phrase tables and two reordering models, MERT didn't converge before the 25th iteration, which is the default maximum number of iterations in Moses. While the log-linear curve seems to be growing higher than the filled-up curves, we consider the more stable behavior of the fill-up models as preferable. Moreover, fast convergence is, by its own, an important property of the fill-up models that allowed us, for example, to tune many other system variants for the IWSLT11 evaluation task with reasonable timings.

5.2. English to French

Our English-French translation experiments closely follow the methodologies described in the Arabic-English task.

Table 4 presents translation quality results in terms of BLEU and NIST scores, using several techniques described in Section 5.1, in addition to the cascaded fill-up model described in Section 3.3. Our first model considers the use of only the TED data for constructing the phrase table and reordering model. In addition to the log-linear and fill-up models experimented on combinations of TED and NEWS data as well as TED and UN data, we combine the three data sources, both with a cascaded fill-up model (*cascade-fill*), and with linear interpolation (*linear*). Due to the erratic behavior of MERT previously observed, we chose not to experiment with log-linear interpolation of more than two phrase and reordering models.

In the cascaded fill-up model, we first construct a fill-up model with the combination of TED and NEWS data and subsequently merge the UN data with the result. Since the size and vocabulary coverage of the UN data is much larger than the NEWS data, we merge the NEWS data first to prevent the UN data from overshadowing the NEWS data. Additionally, the UN data has noise that is not useful in speech translation tasks (for instance, itemized lists). In each fill-up model, we use weights of 0 and 1 for the respective in-domain and background phrase tables, and we only merge phrases that have a source length of 4 or less.

We also include a linear interpolation experiment, in which we provide equal weights for each dataset in both the phrase table and the reordering models.

According to the results in Table 4, the log-linear and linear interpolated translation models and the fill-up models perform virtually the same in terms of BLEU and NIST scores. In the experiments using TED and NEWS data, the log-linear model performs slightly better than the corresponding fill-up model; however the NIST scores report that the fill-up model performs marginally better. It should be noted, however, that these marginal differences are not statistically significant. Likewise, in our experiments with the TED and UN data, we observe that the fill-up model marginally outperforms the log-linear model.

In evaluating the contribution of the NEWS and UN data to the utility of the translation model, we note that the larger vocabulary coverage of the UN data yields higher BLEU scores: a BLEU increase of 0.26 in the log-linear case (not statistically significant) and a 0.4 improvement in the fill-up case (significant at the $p < .01$ level).

Table 4: *%BLEU|NIST scores on English-French TED, using different data combination techniques. All fill-up models use pruning based on newSrcMaxLength=4.*

Translation model	test2010
only TED	29.96 | 7.157
logli(TED+NEWS)*	30.29 | 7.154
fillup(TED+NEWS)	30.22 | 7.177
fillup-n(TED+NEWS)	30.34 | 7.192
logli(TED+UN)*	30.55 | 7.210
fillup(TED+UN)	30.62 | 7.214
cascade-fill(TED+NEWS+UN)	**30.64** | **7.221**
linear(TED+NEWS+UN)	**30.65** | **7.249**

*MERT didn't converge by the 25th iteration.

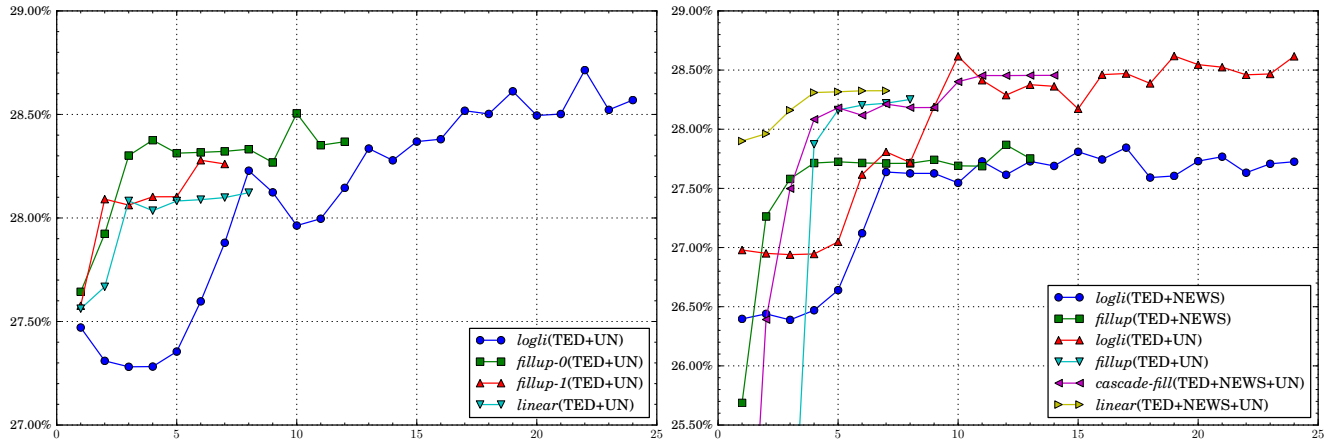

Figure 1: BLEU optimization curves across MERT iterations for the Arabic-English (left) and English-French (right) experiments. *fillup-0* stands for unpruned fill-up, while *fillup-1* stands for *newSrcMaxLength=4*. Scores computed on dev2010.

Our cascaded fill-up model attempted to leverage both the NEWS and UN data; however, the results are only marginally better than experiments that only used the UN data. Likewise, the linear interpolation of the TED, NEWS, and UN data only provided marginal improvements, though again it is not statistically significant.

The significance of the fill-up and linearly interpolated models can be observed in terms of tuning efficiency. As shown in the second graph in Figure 1, the tuning of the fill-up and linearly interpolated models converge much faster than the log-linear combination of phrase and reordering tables. The log-linear models appear to oscillate around the 11th iteration in both the TED+NEWS and TED+UN models and have not converged by the 25th iteration. The corresponding fill-up models converged after the 8th and 13th iterations, respectively. The cascaded fill-up model converged after the 14th iteration. Naturally, were we to have constructed a log-linear model consisting of all three data tables, tuning would not have finished by the 25th iteration.

Interestingly, the linearly interpolated model that assigned equal weights to each phrase and reordering table converged after only seven tuning iterations.

Here too, we compare our phrase penalties with those proposed by [2]. We construct an additional fill-up model using the TED and NEWS data, with phrase penalties 0.5 and 1 (log-space) to indicate in-domain and out-of-domain phrase-pairs, respectively. Using this configuration, MERT does not converge before the 25th iteration – similar to the cases of the log-linear models. In terms of translation quality (see row *fillup-n* in Table 4) we note a marginal improvement over our fill-up configuration, though it is not statistically significant. However, the long number of tuning iterations makes this fill-up model less desirable.

A meaningful excerpt of the cascaded fill-up model is shown in Table 5. We can see how, after fill-up, the polysemous word 'outbreak' gets a much richer set of translations. Also, thanks to the addition of new source phrases not occurring in the TED data, the same word can be expected to be better translated according to its context.

Table 5: *Excerpt of the English-French cascaded fill-up phrase table. Translations coming from each dataset are sorted by phrase translation probability.*

English phrase	French translations
outbreak	*ted:* épidémie / épidémie de / apparition / l'épidémie *news:* poussée *un:* déclenchement / éclatement / début / flambée / explosion / ouverture / éclatent / éruption / ...
contain the outbreak	*news:* contenir l'épidémie *un:* enrayer l'épidémie
latest outbreak	*un:* dernière explosion / nouvelle vague / dernière flambée
outbreak of conflicts	*un:* déclenchement des conflits / éclatement des conflits / éclatement de conflits / apparition de conflits / conflits / ...
outbreak of fire	*un:* existence d'un incendie / début d'un incendie
outbreak of infections	*un:* développement d'infections

As anticipated, the two additional features of the cascaded fill-up model were assigned negative weights during tuning. Specifically, the NEWS and UN features were assigned weights of -0.012 and -0.147, respectively. If interpreted as scaling factors on the translation and reordering models from the NEWS and UN data, the corresponding values are 0.988 and 0.863. According to this configuration,

phrases drawn from the TED data set are most preferable; phrases from the NEWS data set are slightly penalized, but are still reasonable for selection. The UN data, being the furthest in terms of domain from the TED talk task, is assigned a high penalty. While the UN data yields a higher number of candidate phrase translations, the log-linear model marks these phrases as out-of-domain through its higher penalty. The scaling factors also suggest the same order of importance as hypothesized in our construction of the cascade model.

6. Conclusions

We have presented fill-up, an effective data combination technique for phrase-based SMT, and we have systematically evaluated it against standard interpolation methods. Our empirical results corroborate [2]'s conclusion of the overall utility of fill-up models for translation model adaptation.

We have also shown that fill-up models yield comparable results to log-linear combinations of translation models with the additional benefit of efficiency with respect to minimum error training.

When compared to uniformly weighted linear interpolation, fill-up models behave similarly in terms of tuning iterations, and similarly (English-French experiments) or slightly better (Arabic-English) in terms of the reported evaluation metrics. We didn't experiment extensively with mixture weights optimization, therefore we cannot firmly conclude that fill-up is absolutely the best data combination method. However, we know from the literature that mixture weights tuning is still an open problem, and no technique has been shown to strongly outperform the uniformly weighted baseline. On the contrary, fill-up has the advantage of not requiring any tuning procedure other than MERT. For these reasons, we consider fill-up to be an optimal solution for the scenario considered in this paper.

Our implementation of fill-up and cascaded fill-up models will be made available as open-source in the Moses toolkit under the GNU LGPL license.

7. Acknowledgements

This work was supported by the EuroMatrixPlus project (IST-231720), which is funded by the European Commission under the Seventh Framework Programme for Research and Technological Development. We thank the two anonymous reviewers for their helpful suggestions.

8. References

[1] S. Besling and H. Meier, "Language model speaker adaptation," in *Proceedings of the 4th European Conference on Speech Communication and Technology*, vol. 3, Madrid, Spain, 1995, pp. 1755–1758.

[2] P. Nakov, "Improving English-Spanish Statistical Machine Translation: Experiments in Domain Adaptation, Sentence Paraphrasing, Tokenization, and Recasing. ," in *Workshop on Statistical Machine Translation, Association for Computational Linguistics*, 2008.

[3] M. Federico, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2011 Evaluation Campaign," in *International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011.

[4] M. Eck, S. Vogel, and A. Waibel, "Language model adaptation for statistical machine translation based on information retrieval," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, 2004, pp. 327–330.

[5] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," in *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 128–135. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0217

[6] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, E. Hinrichs and D. Roth, Eds., 2003, pp. 160–167. [Online]. Available: http://www.aclweb.org/anthology/P03-1021.pdf

[7] P. Koehn and J. Schroeder, "Experiments in Domain Adaptation for Statistical Machine Translation," in *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 224–227. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0233

[8] K. Yasuda, R. Zhang, H. Yamamoto, and E. Sumita, "Method of Selecting Training Data to Build a Compact and Efficient Translation Model," in *International Joint Conference on Natural Language Processing*, 2008.

[9] A. Axelrod, X. He, and J. Gao, "Domain Adaptation via Pseudo In-Domain Data Selection," in *Conference on Empirical Methods in Natural Language Processing,*, 2011, pp. 355–362.

[10] J. Niehues and A. Waibel, "Domain adaptation in statistical machine translation using factored translation models," in *Proceedings of the 14th Annual Conference of the European Association for Machine Translation (EAMT)*, St. Raphael, France, 2010. [Online]. Available: http://www.mt-archive.info/EAMT-2010-Niehues.pdf

[11] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of HLT-NAACL 2003*, Edmonton, Canada, 2003, pp. 127–133. [Online]. Available: http://aclweb.org/anthology-new/N/N03/N03-1017.pdf

[12] C. Tillmann, "A unigram orientation model for statistical machine translation," in *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2004.

[13] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh system description for the 2005 IWSLT speech translation evaluation," in *Proc. of the International Workshop on Spoken Language Translation*, October 2005.

[14] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Morristown, NJ, USA: Association for Computational Linguistics, 2008, pp. 848–856.

[15] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic, 2007, pp. 177–180. [Online]. Available: http://aclweb.org/anthology-new/P/P07/P07-2045.pdf

[16] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks," in *HLT-NAACL 2004: Short Papers*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 149–152.

[17] H. Johnson, J. Martin, G. Foster, and R. Kuhn, "Improving translation quality by discarding most of the phrasetable," in *In Proceedings of EMNLP-CoNLL 07*, 2007, pp. 967–975.

[18] F. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[19] S. Riezler and J. T. Maxwell, "On some pitfalls in automatic evaluation and significance testing for MT," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 57–64. [Online]. Available: http://www.aclweb.org/anthology/W/W05/W05-0908

[20] E. Hasler, B. Haddow, and P. Koehn, "Margin Infused Relaxed Algorithm for Moses," *The Prague Bulletin of Mathematical Linguistics*, vol. 96, pp. 69–78, 2011.