Phrase Table Induction Using Monolingual Data for Low-Resource Statistical Machine Translation

BENJAMIN MARIE and ATSUSHI FUJITA, National Institute of Information and Communications Technology

We propose a new method for inducing a phrase-based translation model from a pair of unrelated monolingual corpora. Our method is able to deal with phrases of arbitrary length and to find phrase pairs that are useful for statistical machine translation, without requiring large parallel or comparable corpora. First, our method generates phrase pairs through coupling source and target phrases separately collected from respective monolingual data. Then, for each phrase pair, we compute features using the monolingual data and a small quantity of parallel sentences. Finally, incorrect phrase pairs are pruned, and a phrase table is made using the remaining phrase pairs. In our experiments on French–Japanese and Spanish–Japanese translation tasks under low-resource conditions, we observe that incorporating a phrase table induced by our method to the machine translation system leads to large improvements in translation quality. Furthermore, we show that a phrase table induced by our method can also be useful in a wide range of configurations, including configurations where we have already access to large parallel corpora and configurations where only small monolingual corpora are available.

CCS Concepts: • **Computing methodologies → Machine translation**;

Additional Key Words and Phrases: Machine translation, phrase table induction, low-resourced language pairs, semantic similarity, knowledge acquisition

ACM Reference format:
Benjamin Marie and Atsushi Fujita. 2018. Phrase Table Induction Using Monolingual Data for Low-Resource Statistical Machine Translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 17, 3, Article 16 (February 2018), 25 pages.
https://doi.org/10.1145/3168054

1 INTRODUCTION

In statistical and neural machine translation, translation models are usually estimated over a large number of parallel sentences. However, for most language pairs, we do not have such resources, or only in very small quantities, mainly because human translations are costly to produce [13]. Consequently, a state-of-the-art machine translation system cannot be built easily for these language pairs.

Indeed, a translation model trained on a small number of parallel sentences has a poor coverage and a new text to translate is likely to contain many so-called out-of-vocabulary (OOV) tokens or phrases for which the translation model has no candidate translations. Moreover, even for most of

Authors' address: B. Marie and A. Fujita, National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan; emails: {bmarie, atsushi.fujita}@nict.go.jp.

ACM 2375-4699/2018/02-ART16 $15.00
https://doi.org/10.1145/3168054

the tokens seen in the parallel data, the estimated translation probabilities will be unreliable, depending on their frequency. For instance, in phrase-based statistical machine translation (PBSMT) a rare token will have poorly estimated translation probabilities and the translation options registered in the phrase table for this token are likely to be incorrect. The translation model trained on a small number of parallel sentences will thus have relatively accurate translation probabilities only for very frequent tokens and phrases, and the decoder in such conditions will not be able to generate a translation of good quality. In contrast to parallel data, monolingual data are often available in very large quantity for many languages but are usually exploited in PBSMT, for instance, to train a target language model and rarely used to enhance the translation model.

While neural machine translation (NMT) outperforms PBSMT in many configurations, NMT is well-known to produce translations of a much lower quality than PBSMT for low-resource language pairs [25]. Therefore, in this article, we focus on PBSMT. We aim at inducing a phrase table, i.e., a list of phrase pairs associated with features, to improve the translation quality of PBSMT in low-resource conditions by leveraging monolingual data to propose new translation options for unseen or rare phrases in the parallel data. In this work, we assume only the availability of a small number of parallel sentences and a lot of *unrelated* monolingual data.[1] In other words, our work targets the improvement of machine translation for low-resource language pairs involving resource-rich languages. As we review in Section 2, under this condition, none of the existing methods can effectively induce a phrase table containing useful phrase pairs of arbitrary length.

Our method to induce a phrase table is the first one that simultaneously:

- relies on neither large comparable nor parallel data to extract phrases or score phrase pairs,
- extracts and uses a manageable set of phrases of any length to make phrase pairs,
- proposes new useful translation options for both unseen and seen phrases in the parallel data, and
- gives significant improvements of translation quality in very low-resource conditions.

The remainder of this article is organized as follows. We first review previous work in Section 2, highlighting the main advantages and weaknesses of existing methods for inducing a phrase table. Then, in Section 3, we present our phrase table induction method with all the necessary steps: generating phrase pairs using a pair of unrelated monolingual corpora (Section 3.1), computing features for each phrase pair (Section 3.2), and pruning the induced phrase tables to keep their size manageable (Section 3.3). In Section 4, we describe our experiments to evaluate the impact of the induced phrase tables in PBSMT in low-resource conditions. Following the description of the data used in the experiments (Section 4.1), we present our PBSMT systems (Section 4.2) and provide details about the tools and their parameters used to induce the phrase tables (Section 4.3). After making a comparison of state-of-the-art methods with ours (Section 4.4), our main results are given in Section 4.5. Then, in Section 5, we describe further experiments to examine other situations in which our induced phrase table can be useful. We investigate four situations: the availability of large quantity of out-of-domain parallel data (Section 5.1), the use of a pivot language to improve the translation model (Section 5.2), the use of smaller quantities of monolingual data (Section 5.3), and another configuration that does not assume the knowledge of the source side of the development and test sets beforehand as opposed to what previous work has examined (Section 5.4). We present an in-depth analysis in Section 6 to better understand how our induced phrase table is able to help the decoder to generate a better translation. Section 7 draws our conclusions and proposes some possible improvements for our approach.

[1] Unrelated monolingual corpora are corpora in source and target languages for which we do not assume any kind of comparability or parallelism.

2 PREVIOUS WORK

Translating the OOV tokens has been a long-standing issue in the literature of machine translation. The most prominent approach to this issue is bilingual lexicon induction [3, 8, 12, 14, 15, 18, 24, 32]. Irvine and Callison-Burch [18] have proposed an effective method by combining several features to score each candidate pair of words with a classifier and to keep only the pairs with the highest scores to compile a useful bilingual lexicon. They have improved the quality of translations for low-resource language pairs by integrating their induced bilingual lexicon into an SMT system. More recently, Vulić and Moens [40], Han and Bel [15], and Chu and Kurohashi [3] have used word embeddings to generate a bilingual lexicon from monolingual and/or comparable corpora. A completely different trend of work uses an unsupervised method regarding translation as a decipherment problem to learn a bilingual lexicon and uses it as a translation model [10, 29, 33]. However, all the above methods deal only with words, mainly due to the computational complexity of dealing with arbitrary length of phrases. Furthermore, they rely on comparable data and/or find translations for frequent words only.

Induction of phrase tables from monolingual data has been tackled more recently. One of the main shortcomings of the existing methods is that they can only deal with short phrases, up to three words, for instance, due to the computational cost of the methods used. Irvine and Callison-Burch [19] *hallucinate* phrase pairs using all combinations and permutations of unigrams and their translations to obtain short phrases of a length up to three words. This costly approach cannot build phrase pairs that are not compositionally derived from lexical translations. Moreover, it requires the knowledge of the development and test sets beforehand to collect its source phrases and then to generate target phrases given this source phrase set. Saluja et al. [34] and Zhao et al. [42] have built their source phrase set by collecting only unigrams and bigrams from the source side of their development and test data to limit the computational cost of the phrase table induction. They generate new phrase pairs only for unseen source phrases in the parallel data; no new translation option is proposed for seen source phrases. For the target phrases, Saluja et al. [34] have used for a given source phrase a set of target phrases from a baseline system's phrase table that are translations of similar source phrases and additional morphological variants generated by an SMT system or a morphology generator. From this large set of target phrases, they finally keep only the top-r ($r = 20$ in their experiments) candidates according to the forward lexical translation probabilities given by the translation model of their baseline system. Their approach to generate candidate phrase pairs is thus strongly relying on the accuracy of an existing translation model. For instance, if the given source phrase contains only OOV, as it may happen relatively often in low-resource conditions, their approach cannot retrieve candidate target phrases. Zhao et al. [42] have used a simpler method, which collects all unigrams and bigrams in their monolingual corpora that have a probability higher than some threshold according to a language model trained on these data and regards them as the candidate target phrases.

Given a set of candidate phrase pairs, the next step of phrase table induction is to associate each pair with features. The features can then be used to prune the set of phrase pairs and to guide the decoder in using the induced phrase table. Irvine and Callison-Burch [19] score their *hallucinated* phrase pairs using features that may not be available for many language pairs, such as temporal, contextual, and topic similarity features, strongly relying on the comparability of Wikipedia articles and on the availability of news articles annotated with a timestamp [22]. The features are then combined by a classifier to score and rank the phrase pairs to retain only the r-best target phrases for each source phrase. While Saluja et al. [34] have used a costly graph propagation strategy to score the candidate phrase pairs, Zhao et al. [42] have achieved higher BLEU scores with their method that scores and ranks many phrase pairs generated from target phrases, i.e., unigrams and bigrams, collected from monolingual corpora, using only word embeddings, which requires a much lower computational cost. The main contribution of Zhao et al. [42] is the use of a local linear projection (LLP) strategy, instead of the commonly used single global linear projection (GLP), to estimate the semantic similarity for each phrase pair. It makes the projection of the source embeddings to the target embedding space by learning a translation matrix for each source phrase embedding, trained on gold phrase pairs with source phrase embeddings similar to the one to project. For each projected source phrase, based only on the similarity over embeddings, the r-nearest target phrases are retrieved. If the projection for a given source phrase is not accurate enough, then very noisy phrase pairs are generated. This may happen especially if we only use a small number of parallel sentences to learn the so-called gold, but mostly incorrect, phrase pairs used to train the translation matrices (see Sections 4.4 and 4.5 for empirical evidences). This may also be a problem when the given source phrase does not need to be translated (i.e., numbers, dates, name entities, etc.). The system will recklessly translate the source phrase, because it is originally OOV but now registered in the induced phrase table, but with only wrong translations.

In summary, all of the existing methods for inducing a phrase table rely strongly on resources that may not be available for low-resource language pairs: comparable corpora at document level [19] or sentence level [17], comparable phrase sets [9], or an accurate translation model trained on large parallel corpora [34, 42].

It is worth mentioning that there are several studies that have proposed new features to score phrase pairs in a given phrase table either by using comparable data [22] or word embeddings trained on large parallel data [31]. These are orthogonal to phrase table induction, as they merely provide new features for existing phrase pairs.

3 PHRASE TABLE INDUCTION

In our proposed method for phrase table induction, we combine advantages of the previous work presented in Section 2, while alleviating some of their limitations. Unlike existing methods that assume comparability, at document level [19], sentence level [17], or parallelism [34, 42] of their source and target data, for efficiency, or for computing accurate features, our method assumes that neither comparable corpora nor large parallel corpora are available. We use a large quantity of unrelated source and target monolingual data, which is available for many languages, on the Web, for instance, and a small quantity of parallel data, which can be acquired through crowdsourcing, for instance. Futhermore, as opposed to previous work that collects only unigrams or bigrams [34, 42], or short phrase pairs generated from word-level translations [19], we collect pairs of phrases of arbitrary length. Finally, our method computes different kinds of features for each phrase pair, instead of relying only on word embeddings [42] or features that can be computed efficiently when document pairs are available [19, 22]. The resulting set of phrase pairs obtained with associated features is then used as a phrase table that we jointly use with other existing models in a PBSMT system.

In this section, we describe our three-step procedure to induce a phrase table. We first present how to identify and collect phrases for the source and target languages to generate candidate phrase pairs (Section 3.1). Then, we detail the features used to evaluate the likelihood of each pair of source and target phrases to be translations (Section 3.2). Finally, to retain only the useful phrase pairs in the final induced phrase table, we present a method to assess, rank, and filter the phrase pairs using a binary classifier (Section 3.3).

3.1 Candidate Phrase Pairs Generation

First, our procedure extracts phrases from monolingual data. In a standard configuration, PBSMT systems extract phrases of a length up to six or seven words. Although it is feasible to collect all the

n-grams of such a length from given large monolingual corpora, it will provide a large set of source and target phrases, resulting in an enormous number of candidate phrase pairs. In contrast with previous work, we collect more meaningful phrases than arbitrary short n-grams from source and target monolingual data, independently. To collect meaningful phrases from large monolingual corpora, we need a computationally efficient method that collects phrases composed of words that are not necessarily of high frequency. For this purpose, we choose to use the formula proposed by Mikolov et al. [27], similar in spirit to point-wise mutual information but using discounted frequencies. It identifies a sequence of two tokens as a phrase if the two tokens appear frequently together relatively to their individual frequency in the monolingual data:

$$score(w_i w_j) = \frac{freq(w_i w_j) - \delta}{freq(w_i) \times freq(w_j)},$$

where w_i and w_j are two consecutive words or phrases in the monolingual data, $freq(\cdot)$ the frequency of the given word or phrase, and δ a discounting coefficient for preventing the retrieval of phrases composed from very infrequent words. All the bigrams $w_i w_j$ in the given monolingual corpus are scored by this formula, and only the bigrams with a score above a predefined threshold θ are considered as phrases. A new pass is performed to obtain longer phrases, regarding the identified phrases in the previous pass as one word.[2] After several passes, we retain the original words and phrases identified in each pass that appear at least K times in the monolingual data. This frequency-based constraint also prevents us from collecting phrases for which we will not be able to compute reliable features, including the one based on word embeddings (see Section 3.2.1). The retrieved set of phrases contains the single words and all the phrases with a length of up to L words identified in each pass and after T passes.

Given the two sets of phrases, their Cartesian product is regarded as the set of candidate phrase pairs. This exhaustive exploration enables us to find phrase pairs that are compositionally and not compositionally derived from lexical translations.

3.2 Feature Computation

We now have a large set of phrase pairs and need to score each of them to identify truly relevant ones. This section presents features to characterize the likelihood of each pair of source and target phrases to be translations. Such features must be efficient to compute and useful to discard irrelevant phrase pairs. They must also be useful for the decoder to effectively exploit the induced phrase table. We use the following features, since they have provided us satisfying results in our preliminary experiments, but many more features can be incorporated in our method.

3.2.1 Cross-Lingual Semantic Similarity. Recently, many researchers tackled the problem of estimating semantic similarity between pairs of words or phrases in two different languages, using word or phrase embeddings [1, 6, 11, 41]. To estimate this cross-lingual semantic similarity for each phrase pair, we first estimate monolingual phrase embeddings for each of our collected phrases. Given the monolingual data used to collect the phrases, we train word embeddings for each word and obtain phrase embeddings through the element-wise addition of the embeddings of constituent words of the phrase. We adopt this method because it performs well to estimate phrase embeddings [26, 28] with a relatively low computational cost. Nonetheless, our method is agnostic to the choice of the phrase representation. More costly state-of-the-art methods based on neural networks [35, 36] may give better results.

The source phrase embeddings and the target phrase embeddings are in two different monolingual embedding spaces. We then project linearly the source phrase embeddings to the target

[2] This transformation is performed by simply replacing the space between the two words/phrases with an underscore.

embedding space. To perform this projection, we choose the method proposed by Mikolov et al. [26] considering its low computational cost and its reasonable need of external resources to estimate the translation matrix for the projection. More accurate methods need comparable or parallel data [37] in a large quantity that we cannot assume for low-resource language pairs. Given training data, i.e., a small gold bilingual phrase lexicon and the corresponding pairs of source and target phrase embeddings, a translation matrix $\hat{W}$ is computed by solving the following convex optimization problem with stochastic gradient descent:

$$\hat{W} = \arg\min_{W} \sum_{i} ||Wx_i - z_i||^2,$$

where x_i is the source phrase embedding of the i-th training data, z_i the target phrase embedding of the corresponding gold translation, and W the translation matrix used to project x_i such that Wx_i is as close as possible to z_i in the target embedding space. One important parameter here is the number of dimensions for word and phrase embeddings. Those for the source and target embeddings can be different, but must be smaller than the number of phrase pairs in the training data; otherwise, the equation is not solvable. Section 4.1 provides details about the lexicons that we have used in our experiments.

The projection is then performed using $\hat{W}$ to project all the source phrase embeddings to the target embedding space. As shown by Mikolov et al. [26], the projection is more accurate when using a higher number of dimensions for the source word embeddings, D_e, than that for the target word embeddings, D_f. Therefore, we train a translation matrix for each translation direction, i.e., $f \rightarrow e$, with $D_f > D_e$, and $e \rightarrow f$, with $D_e > D_f$, to obtain two cross-lingual semantic similarity features for each phrase pair; given two phrase embeddings, we compute their cosine similarity.

3.2.2 Lexical Translation Probabilities. We assume that even for low-resource language pairs it is possible to collect a small amount of sentence pairs with some effort. Using such parallel data, we can train a basic translation model. Although such a model will have a very low coverage and poorly estimated translation probabilities, the estimates for very frequent tokens may be useful to evaluate the similarity between phrases that contain such frequent tokens.

To compute a translation score at phrase level, for a target phrase e given a source phrase f, we use the following formula considering all possible word alignments and their corresponding lexical translation probability:

$$P_{lex}(e|f) = \frac{1}{I}\sum_{i=1}^{I} \log\left(\frac{1}{J}\sum_{j=1}^{J} p(e_i|f_j)\right),$$

where I and J are the number of words in the target and source phrases, respectively, and $p(e_i|f_j)$ the lexical translation probability of the i-th target word e_i given the j-th source word f_j. The phrase-level lexical translation probabilities are estimated for both translation directions giving us two features for each phrase pair.

3.2.3 Phrase Frequency and Phrase Length. As demonstrated by previous work [19, 20], features based on the frequency of the phrases in the monolingual data may help to better estimate the similarity between two phrases. Indeed, we can expect that words or phrases that are translation of one another have a similar relative frequency in their respective language. Therefore, we add as features the inversed frequency of the source and target phrases in the given monolingual data, along with the absolute value of the difference between the log of their relative frequencies, given by the following formula:

$$sim_f(e, f) = \left|\log\left(\frac{freq(e)}{N_e}\right) - \log\left(\frac{freq(f)}{N_f}\right)\right|,$$

where N_x stands for the number of words in the monolingual data of the corresponding language.

We also add as features the phrase lengths, i.e., I and J, and their ratio.

3.3 Phrase Pair Filtering

Pruning inappropriate phrase pairs is the final step of our phrase table induction. This pruning aims at limiting the size of the phrase table and, consequently, the size of the decoder's search space when using the induced phrase table.

For this purpose, we introduce a binary classifier to predict how likely phrases in a phrase pair are translations of one another, as proposed by Irvine and Callison-Burch [18]. The classifier is used to give a real-valued confidence score for each phrase pair. Then, for each source phrase, the k-best target phrases, according to their score, are kept as possible translations. To train the classifier, we regard gold translations in a given bilingual phrase lexicon as positive examples and randomly paired source and target phrases from our source and target phrase sets as negative examples. As the features for classification, we use all those presented in Section 3.2. For decoding, the confidence score estimated by the classifier is also added as a feature in the induced phrase table.

Furthermore, since we may extract the source phrases directly from the text to translate, it is likely that our translation system will encounter words to translate that are unseen in the monolingual data. In this case, we will not have, for instance, any embeddings associated with such words; this is problematic to compute the features presented in Section 3.2.1. To avoid this situation, we do not retain phrase pairs containing words unseen in the monolingual data.

4 EXPERIMENTS

To evaluate the impact on translation quality of using a phrase table induced with our method, we conducted experiments with the language pairs French–Japanese (Fr–Ja) and Spanish–Japanese (Es–Ja) for both translation directions. We chose these language pairs as they are low resource, with only a small number of publicly available parallel sentences, while they are widely spoken, with a large quantity of monolingual data publicly available. Note that we were not able to apply most of the other existing methods for inducing phrase tables, such as those reviewed in Section 2, for comparison purposes, because we did not exploit large comparable data, but used only a very small amount of parallel data for most of our configurations. The only methods we were able to apply with our data are the GLP and LLP methods proposed by Zhao et al. [42].

Section 4.1 describes the data used to build the SMT systems and to induce the phrase tables. Section 4.2 presents the SMT systems and the models, and Section 4.3 presents the tools and their parameters used to induce the phrase tables. In Section 4.4, we present the results obtained with the GLP and LLP methods [42], followed by Section 4.5, in which we discuss the results obtained with our method on two different translation tasks.

4.1 Data

We performed experiments on two translation tasks, denoted A and B, to investigate the impact of our method on translation quality, using several different sizes of parallel data to train the basic phrase table (henceforth, bpt) and to obtain our induced phrase table (henceforth, ipt). For

Table 1. Statistics of Bilingual Data Used to Train the Phrase Tables

		Corpus	#sentences	#tokens		
				Ja	Fr	Es
Ja-Fr	A	development	1,000	11k	10k	
		test	3,000	33k	28k	
		Tatoeba	27,187	300k	256k	
	B	development	2,909	50k	48k	
		test	2,871	49k	48k	
		P1	334,719	5.7M	5.5M	
Ja-Es	A	development	1,000	11k		8k
		test	3,000	31k		25k
		Tatoeba	23,985	248k		186k
	B	development	2,880	49k		44k
		test	2,836	49k		43k
		P1	331,938	5.6M		5.0M

task A, we used the publicly available Tatoeba parallel corpus.[3] The development and test sets were extracted from Tatoeba and the bpt was trained on 5k, 10k, or all the sentence pairs. For task B, the development and test sets were extracted from an in-house parallel corpus of daily life conversations, denoted P1, and the bpt was trained on 5k, 10k, 20k, 40k, 80k, 160k, or all the sentence pairs. For each configuration, we trained bpt and ipt, and generated plots presenting BLEU scores obtained with the SMT systems using these phrase tables. We can expect the systems trained for task B to be more difficult to improve than the systems trained for task A. Indeed, for each task the parallel data used by the systems are of a very different nature. The parallel data used to train, tune, and test the systems for task A can be very heterogeneous as they may refer to various domains and are created via crowdsourcing by many different, presumably, non-professional translators. Development and test data can then be different to some extent from the training data, with a lot of OOV words and phrases, for instance, that our ipt can help to translate. In contrast, the parallel data used to train, tune, and test the systems for task B are very homogeneous, as they are all from the same domain and created by a limited number of professional translators. Therefore, since the bpt is trained on data very similar to the development and test data, our ipt may have less impact in the systems trained for this task.

We used completely unrelated source and target monolingual data to perform the induction of our phrase tables. Our French and Spanish monolingual data comprise Europarl and all the News corpora provided by WMT'15 for French[4] and WMT'13 for Spanish,[5] while our Japanese monolingual data are composed of excerpts from Web-crawled Japanese documents. French and Spanish data were preprocessed using the tokenizer provided by moses [23],[6] while Japanese data were tokenized using MeCab.[7] The statistics of the bilingual and monolingual data used for our experiments are presented in Table 1 and Table 2, respectively.

We also need bilingual phrase lexicons to compute the translation matrix used to project phrase embeddings (see Section 3.2.1) and to train the binary classifier used to score the phrase pairs (see

[3] https://tatoeba.org/eng/; downloaded from http://opus.lingfil.uu.se/Tatoeba.php.
[4] http://statmt.org/wmt15/translation-task.html.
[5] http://statmt.org/wmt13/translation-task.html.
[6] http://www.statmt.org/moses/.
[7] https://github.com/taku910/mecab/.

Table 2. Statistics of the Monolingual Data Used to Train the Language Models and Induce the Phrase Tables

Language	#lines	#tokens
Ja	40M	1.0B
Fr	44M	1.1B
Es	16M	446M

Section 3.3). Having obtained satisfying results during preliminary experiments, we choose the same bilingual phrase lexicon for both purposes. We extracted this lexicon from the bpt. As we trained the bpt using only a small quantity of parallel data, we generated the lexicon by extracting only the 2k most frequent source phrases and their most probable translation, according to the forward translation probability given by the bpt. We considered that this size of lexicon may be enough to train the translation matrix [39] and the classifier, and left the exploitation of more and consequently less accurate phrase pairs for future work.

4.2 SMT System

We compared two SMT systems: one baseline system using only the bpt, trained on parallel sentences using mgiza, and the other one using the same bpt and our ipt. We used the moses toolkit [23] to build and test all the systems. Our systems used up to two phrase tables, using the multiple decoding path ability of moses.[8] All the systems used one msd-bidirectional-fe lexical reordering model trained on the parallel data and two 4-gram language models: one trained using lmplz [16] on the entire monolingual data of the target language and the other trained on the target side of the parallel data used to train the bpt.

The weights of the models and features were optimized using kb-mira [2] on the 200-best translation hypotheses through 15 iterations. The translation outputs were evaluated with BLEU [30], and the scores were averaged over three tuning runs. The statistical significance was measured by approximate randomization [4] using MultEval.[9]

4.3 Settings for Phrase Table Induction

As in previous work [19, 34, 42], we computed only the phrase pairs whose source side appears in the development and test sets (henceforce, devtest sets) to maximize the coverage of the ipt. This is the ideal scenario for which we know the devtest sets beforehand; we analyze in Section 5.4 the impact of this choice. Source phrases were thus extracted from the concatenation of the source side of the devtest sets and 1M lines randomly sampled from the monolingual data. On the other hand, target phrases were extracted from the concatenation of the monolingual data and the target side of the parallel data. Phrases were extracted through four passes of word2phrase, a tool provided in the word2vec package,[10] with its default values for the parameters δ and θ, and a maximal phrase length of 6. To limit the number of phrases, we considered only the words and phrases appearing in the monolingual data at least K times: $K = 5$ for source phrases and $K = 100$ for target phrases. As a result, we obtained between 10k and 24k source phrases that are appearing in the source side

[8] We used the "either" strategy to use multiple phrase tables during decoding with moses. If a phrase pair appears in more than one phrase table, then different decoding paths are created and each considers only the corresponding features for scoring.

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

[10] http://word2vec.googlecode.com/.

Table 3. Number of Source Phrases Extracted for the Given Devtest Sets Using the Concatenation of Them and Source Monolingual Data

Translation direction		#source phrases	
		Task A	Task B
(a)	Fr→Ja	11k	24k
(b)	Ja→Fr	10k	22k
(c)	Es→Ja	10k	21k
(d)	Ja→Es	10k	22k

Table 4. Number of Target Phrases Extracted from the Target Monolingual Data Concatenated to the Target Side of Different Sizes of Parallel Data Used to Train the bpt. The Second Row Shows the Number of Sentence Pairs in the Parallel Data

Language	Task A			Task B						
	5k	10k	all	5k	10k	20k	40k	80k	160k	all
Ja	385k	385k	385k	385k	385k	385k	385k	385k	386k	386k
Fr	393k	393k	393k	393k	393k	393k	393k	393k	394k	395k
Es	271k	271k	271k	271k	271k	271k	271k	272k	272k	273k

Table 5. Software Parameters

`word2vec`	-cbow 1 -window 10 -negative 15 -sample 1e-4 -iter 15 -min-count 5
`Vowpal Wabbit`	–passes 1 –normalized –loss_function logistic –link logistic

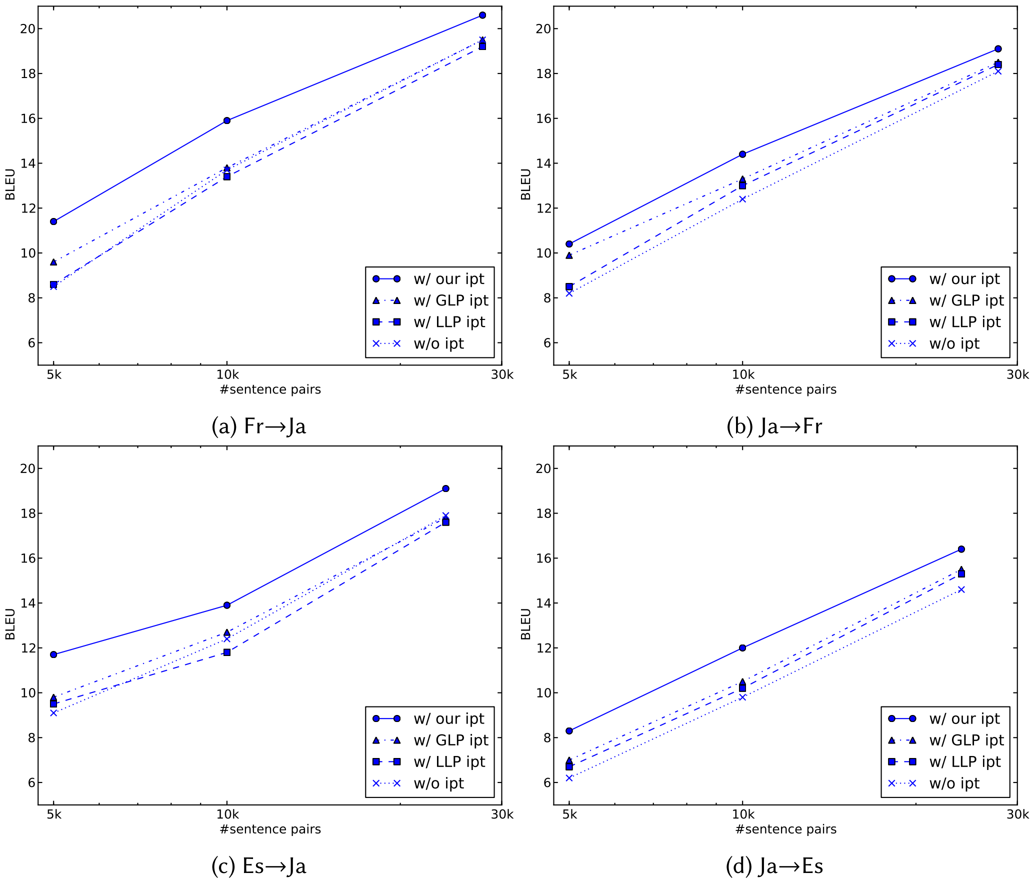
of the devtest sets, and between 271k and 395k target phrases. See Tables 3 and 4 for details for each configuration.

For each configuration, word embeddings were learned from the concatenation of the monolingual data and the corresponding side of the parallel data using `word2vec`; we used a different number of dimensions for the source and target sides, respectively, 800 and 300, following Mikolov et al. [26]. Each phrase pair was scored using the linear classifier `Vowpal Wabbit`[11] on the basis of the features described in Section 3.2. Finally, for each source phrase, we kept the 300-best target phrases according to the classifier's score.[12] The parameters for the aforementioned tools are given in Table 5; we did not try to tune them.

To compare our work with a state-of-the-art phrase table induction method, we implemented the work of Zhao et al. [42]. Even though they did exploit parallel data much larger than ours, their work is the closest to ours; it does not require other external resources than those we used, i.e., parallel data and unrelated monolingual data. We implemented both global (GLP) and local (LLP) linear projection strategies to induce phrase tables with our phrase sets. To get the best possible results, we did not use the search approximations presented in Zhao et al. [42], i.e., local sensitive hashing and redundant bit vector, and instead used linear search. For the GLP configurations, the translation matrix was trained on the same 2k phrase pairs used by our method (see Section 4.1). For the LLP configurations, as in Zhao et al. [42], we trained the translation matrix for each source

[11] http://hunch.net/~vw/.

[12] Favoring recall over precision, with a larger number of target phrases, gives better results. However, we observed no more improvements by keeping more than the 300-best target phrases.



(a) Fr⟶Ja

(b) Ja⟶Fr

(c) Es⟶Ja

(d) Ja⟶Es

Fig. 1. Results (BLEU) for task A for different configurations: without using an ipt or using an ipt induced by GLP, LLP, or our method.

phrase on the 500 most similar source phrases, retrieved from the bpt, associated with their most probable translation. For both GLP and LLP configurations, we kept the 300–best target phrases for each source phrase. Four features for each phrase pair, i.e., phrase and lexical translation probabilities for both translation directions, were approximated on the basis of the similarity between (projected) source and target phrase embeddings and included in our ipt as described by Zhao et al. [42].

4.4 Preliminary Experiments with GLP and LLP Induction Methods

The quality of the translations obtained using GLP or LLP ipt for task A are presented in Figure 1. For nearly all the configurations and translation directions, the systems using a GLP ipt performed significantly better than the system that do not use any ipt. The largest gain, +1.7 BLEU points, was observed in the configurations using only 5k parallel sentences for the Ja⟶Fr translation direction. The improvements tend to be reduced according to the increase of the number of parallel sentences used. In contrast, using the LLP ipt led to a drop of the BLEU score compared to the system that do not use any ipt, for the configurations using a small number of parallel sentences for the Fr⟶Ja and Es⟶Ja translation directions. These results seem to be in contradiction to the results presented by Zhao et al. [42], i.e., a better performance with LLP than with GLP. We can explain such a difference by the use of a very limited amount of parallel data, since LLP needs many accurate gold phrase pairs in the bpt to train many translation matrices. However, only few accurate phrase pairs can be retrieved when a bpt has been trained only on a limited number of sentence pairs. Consequently, the translation matrices trained on very noisy bilingual phrase lexicons are likely to be inaccurate, leading to the ipt of poor quality. As we can see, using all the sentence pairs from Tatoeba gives better performances with LLP and brings it very close to the performance obtained with GLP.

An interesting property of the GLP and LLP methods is their relatively low computational cost. For instance, inducing phrase tables for the Fr→Ja translation direction required 24 min for GLP and 35 min for LLP. On the other hand, the induction with our method for this translation direction was much slower: It spent 3 h 52 min.[13]

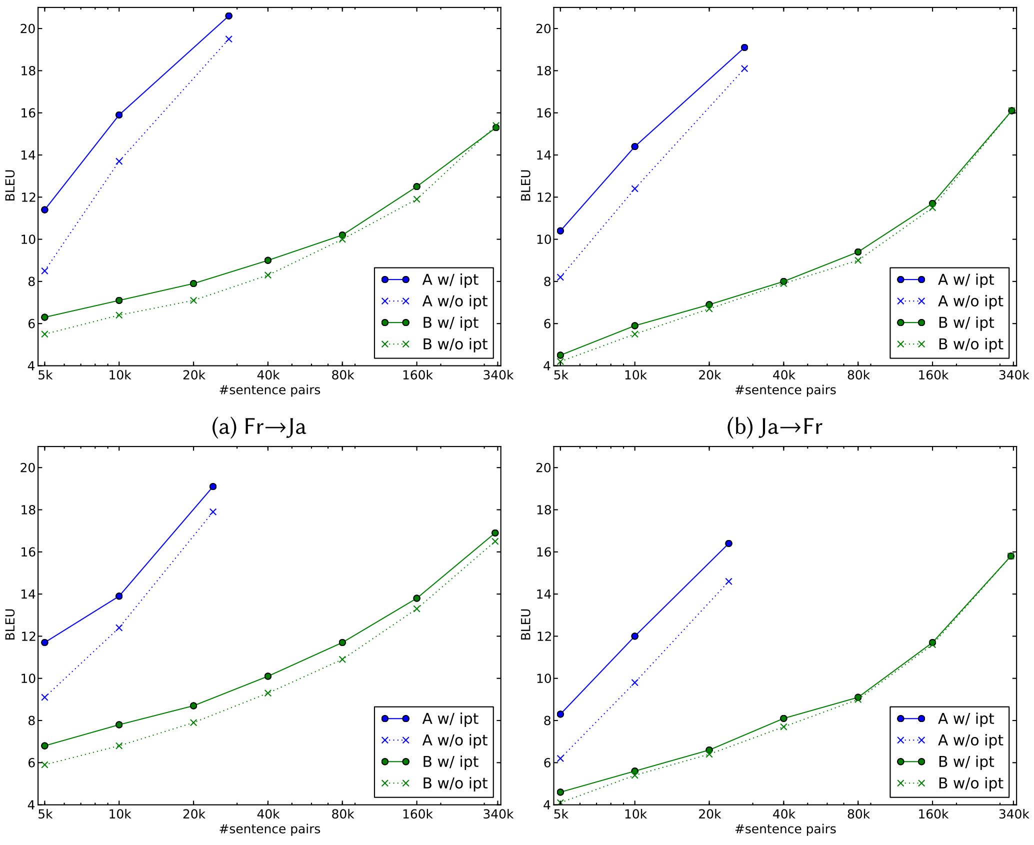
Nonetheless, with such a small quantity of parallel data, GLP and LLP induction methods seem inadequate and may only give similar or worse results than a configuration that does not use any ipt. Furthermore, as pointed out by Zhao et al. [42], the performance of these induction methods strongly depends on the quality of the word embeddings. We can only expect worse results for low-resource languages with a limited amount of monolingual data available to train the word embeddings. Unlike GLP and LLP, the systems using our ipt brought significantly better results for all the configurations and translation directions. Henceforth, we discuss and analyze only our ipt, jointly with the results obtained for task B.

Our results for both translations tasks A and B are presented in Figure 2. First, for nearly all the configurations and translation directions, adding our ipt improved significantly the translation quality evaluated with BLEU. Especially in very low-resource conditions, Es→Ja systems trained on 5k sentence pairs for task A, for instance, adding an ipt improved the BLEU score from 9.1 to 11.7 (+2.6 BLEU points). We observed similar improvements for the other translation directions. The improvements tended to be reduced when more sentence pairs were used to train the bpt. For instance, the BLEU score was increased from 17.9 to 19.1 (+1.2 BLEU points) by adding the ipt to the Es→Ja system using all the Tatoeba sentence pairs. The same tendency was observed for task B, with smaller but still significant improvements for the Fr→Ja and Es→Ja translation directions. For instance, with the systems using 20k sentence pairs, adding the ipt brought 0.8 BLEU points of improvements for Fr→Ja and Es→Ja. Using all the sentence pairs in P1, we still obtained 0.4 BLEU points of improvement for Es→Ja, but no more significant improvements for the other translation directions. These reduced improvements can be explained by the much better quality of the bpt trained on more parallel data, which proposes more accurate translations and reduces the number of OOV tokens making our ipt less effective.

4.5 Results

We can also notice that the improvements brought by our ipt are smaller for task B than for task A, especially for the Ja→Fr and Ja→Es translation directions. We expected these differences (see Section 4.1). The data used to train, tune, and test the systems are more homogeneous for B than for A, meaning that the bpt fits more the devtest sets with a better coverage and more accurate translations reducing the impact of our ipt.

[13] The experiments were performed with 20 CPU threads. The reported times do not include the phrase collection step that is commonly required for all the methods. For instance, it took 47 min for French on a single CPU thread. Note also that computational speed was not our primary focus when implementing our method. Optimizing our implementation may lead to significant gains in speed. Zhao et al. [42] made their approach up to 18 times faster than linear search by using search approximations.



(a) Fr→Ja

(b) Ja→Fr

(c) Es→Ja

(d) Ja→Es

Fig. 2. Results (BLEU) for tasks A (blue curves) and B (green curves) with regard to the number of sentence pairs used to induce our ipt.

Our experiments also revealed that our ipt can bring as much improvement as adding twice more sentence pairs to train the bpt. For instance, on task B for the Fr→Ja and Es→Ja translation directions, using our ipt and a bpt trained on 10k sentence pairs achieved almost the same performance as using the single bpt trained on 20k sentence pairs. This result highlights the potential of our ipt induced from monolingual data to alleviate the costly need of producing more parallel data, at least in low-resource condition, to achieve a similar translation quality.

5 ADDITIONAL EXPERIMENTS IN DIFFERENT CONDITIONS

In Section 4, we have demonstrated the usefulness of our ipt in a very favorable but also realistic scenario for our approach. More precisely, we have used a large quantity of monolingual data to induce a reliable phrase table and only a small quantity of parallel data to train a noisy bpt that was the only phrase table used by the baseline system. While we can assume the availability of these resources for many language pairs, for the other many language pairs, we may have access to more parallel data, directly involving the source and target languages or indirectly through pivot languages. Another possible situation is the unavailability of large quantity of monolingual data, especially when we deal with low-resource languages.

Table 6. Results (BLEU), Where the First Two Rows Correspond to the Rightmost Points of the Task A Lines: All the Configurations Use bpt Trained on the Entire Tatoeba, with or without Our Induced Phrase Table (ipt) and the Phrase Table Trained on the Out-of-domain Corpus (P2)

Plugged phrase tables			Fr→Ja	Ja→Fr	Es→Ja	Ja→Es
Tatoeba	Tatoeba + P2	ipt				
√			19.5	18.1	17.9	14.6
√		√	20.6	19.1	19.1	16.4
	√		20.7	23.0	19.4	20.0
	√	√	**21.4**	**23.4**	**19.9**	20.2

Bold scores indicate statistical significance ($p < 0.01$) of the score over the baseline system using only the bpt trained on Tatoeba+P2.

In this section, we investigate the impact of our ipt in each of these situations using task A (see Section 4). First, in Section 5.1, we use a much larger quantity of parallel data from an out-of-domain corpus to train the bpt and to induce the ipt. Then, in Section 5.2, we resort to a pivot language to build a new phrase table and evaluate its impact on the usefulness of our ipt. In Section 5.3, we investigate the impact of the quantity of the monolingual data used to induce the ipt. Finally, in Section 5.4, we examine how the use of the devtest sets affects the phrase collection step of our method and the effectiveness of our ipt.

5.1 With a Large Quantity of Out-of-Domain Parallel Data

In some situations, we may have access to a large quantity of out-of-domain parallel data in addition to the smaller quantity of in-domain parallel data that is usually used to build an in-domain SMT system. We experimented this scenario by using a large in-house out-of-domain parallel corpus of traveling expressions, denoted P2 (465k sentence pairs), in addition to the entire Tatoeba corpus. In this experiment, we used three different phrase tables and our SMT systems jointly used up to two of them:

Tatoeba A bpt trained on the entire Tatoeba

Tatoeba+P2 Another bpt trained on P2 concatenated[14] to the entire Tatoeba

ipt An ipt induced using the entire source and target monolingual data concatenated to the corresponding sides of the Tatoeba and P2 parallel corpora

Compared to our experiments presented in Section 4, we expect that the baseline system, additionally using P2, will lead to better results, because it uses a lot more parallel data to train a phrase table with better estimated translation probabilities.

Our results are presented in Table 6. As shown in the third row, introducing the Tatoeba +P2 phrase table significantly improved the translation quality of the system that uses only the Tatoeba phrase table. These improvements are largely due to the reduction of the number of OOV and the use of translation probabilities better estimated on the large number of parallel sentences in P2. The improvements are much more remarkable for the language pairs with Japanese as a source language, +4.9 and +5.4 BLEU points for Ja→Fr and Ja→Es, respectively, which are much larger than the impact of our ipt. On the other hand, the gains of +1.2 and +1.5 BLEU points for Fr→Ja and Es→Ja, respectively, are only slightly larger than those obtained with our ipt. Nonetheless,

[14] Training one single bpt from Tatoeba concatenated to P2 gave the best results among several configurations to use P2, including the use of two phrase tables trained separately on Tatoeba and P2.

Table 7. Statistics on All the Parallel Data Used for Pivot-Based Systems

Corpus	Sentences	Tokens			
		Ja	En	Fr	Es
Europarl	2M	-	56M	62M	-
	2M	-	55M	-	57M
in-house Ja-En	2M	42M	34M	-	-
	26M	538M	435M	-	-

as shown in the last row, adding the `ipt` to the system using the `Tatoeba+P2 bpt` consistently brought significant improvements, with the largest improvement of +0.7 BLEU point for the Fr→Ja translation direction. This highlights that even when a lot of out-of-domain data are available to train a new `bpt`, our `ipt` is still useful to further improve the SMT system.

5.2 With a Pivot Language

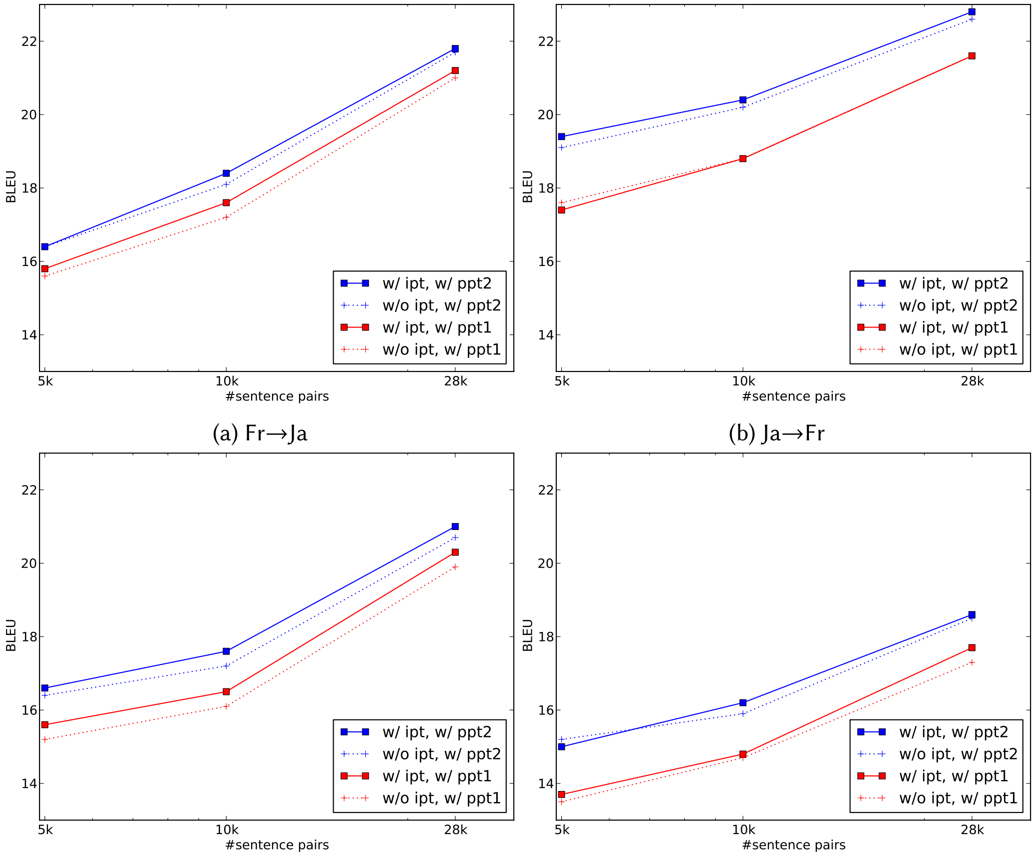
Using a pivot-language is a well-known alternative to circumvent the lack of parallel data in dealing with low-resource language pairs. We examined the impact of our `ipt` on this method, using English–French and English–Spanish versions of the Europarl Parallel Corpus[15] (2M sentence pairs) and an in-house English–Japanese parallel corpus (26M sentence pairs) created from various sources, including manual translations of Wikipedia's Kyoto articles[16] and automatically aligned patent data but no sentence pairs from `Tatoeba`. Note that this is an ideal situation for a pivot-based configuration, as we used a very large amount of parallel data. However, this situation is relatively uncommon as we may not have such a large quantity of parallel data with a pivot language for other low-resource language pairs. Therefore, to evaluate the impact of our `ipt` under a condition with a smaller parallel corpus, we also created a smaller English–Japanese parallel corpus by randomly sampling 2M sentences. The French, Spanish, and Japanese data were preprocessed in the same manner as in Section 4.1. English data were preprocessed using the tokenizer provided by `moses`, as for French and Spanish. Statistics of the resulting corpora are summarized in Table 7.

Our pivot-based SMT systems were built using the following procedure. First, a total of six PB-SMT systems, i.e., {Fr,Es,Ja}→En and En→{Fr,Es,Ja}, were separately trained on the corresponding parallel corpus. Significance pruning [21] was performed to filter out relatively unreliable phrase pairs. Then, new phrase tables for {Fr,Es}→Ja and Ja→{Fr,Es} translations were generated by the triangulation method [5, 38], regarding English as the pivot language. Although we had pruned the component phrase tables, this triangulation generated a large number of new phrase pairs including those that were totally nonsensical. To reduce the computational cost for decoding as well as the negative effects potentially caused by this noise, for each source phrase s, we retained only the 40-best translations t on the basis of $\phi(t|s) = \sum_{p \in E_{SP} \cap E_{PT}} \phi(t|p)\phi(p|s)$, i.e., the forward translation probability calculated from those in the component models, where E_{SP} and E_{PT} stand for the set of English phrases in source-pivot and pivot-target language pairs, respectively. For each of the retained phrase pairs, we also added the backward translation probability, $\phi(s|t)$, and lexical translation probabilities, $\phi_{lex}(t|s)$ and $\phi_{lex}(s|t)$, in the same manner as $\phi(t|s)$.

We evaluated the impact of introducing a pivot-based phrase table in combination with our `ipt`, i.e., we used up to three phrase tables: `bpt`, pivot-based one, and our `ipt`. Figure 3 presents the results using 5k, 10k, or all the sentence pairs from `Tatoeba` to train a `bpt` and to induce our

[15] http://statmt.org/europarl/, release 7.

[16] http://alaginrc.nict.go.jp/WikiCorpus/.



(a) Fr→Ja

(b) Ja→Fr

(c) Es→Ja

(d) Ja→Es

Fig. 3. Results (BLEU) of integrating a pivot-based phrase table into the SMT system.

ipt, as in Section 4. The pivot-based phrase tables are denoted ppt1 and ppt2, respectively, built using 2M or 26M Ja-En parallel sentences. As expected, the systems using the pivot-based phrase tables achieved much higher scores, thanks to the translation model better estimated through the pivot language. However, although adding our ipt on top of the pivot-based phrase tables constantly brought improvements, these improvements were not significant when using ppt2, while they were significant for some configurations when using ppt1 but never higher than +0.4 BLEU points. This is the consequence of the use of a very large amount of data to build the pivot-based phrase tables. The translation probabilities computed for their phrase pairs are relatively accurate compared to the features in the ipt that are mainly derived from monolingual data. Moreover, as shown in Table 8, since we used a lot of parallel data to build the pivot-based phrase tables, their coverage was much higher than that of the bpt and closer to the coverage of the ipt while proposing more accurate translations. Indeed, for the Ja→Fr translation direction, for instance, the source side of the test set contained 1,817 OOV tokens when using only the bpt trained on 5k sentence pairs. The number of OOV tokens significantly decreased to 249 when using ppt1 or 53 when using ppt2 which is very close to the 41 OOV tokens attained when using our ipt.

Table 8. Number of OOV Tokens in the Test Set Depending on the Phrase Tables Used by the Decoder

Plugged phrase tables							
bpt	ppt1	ppt2	ipt	Fr→Ja	Ja→Fr	Es→Ja	Ja→Es
√				1,788	1,817	3,086	2,731
√	√			330	249	665	319
√		√		318	53	636	70
√			√	57	41	104	24

Both of the bpt and the ipt were built using 5k sentence pairs from Tatoeba.

Since our ipt is more helpful to the systems with ppt1 than to those with ppt2, we assume that our ipt will remain more helpful to the configurations with pivot-based phrase tables built from less parallel data.

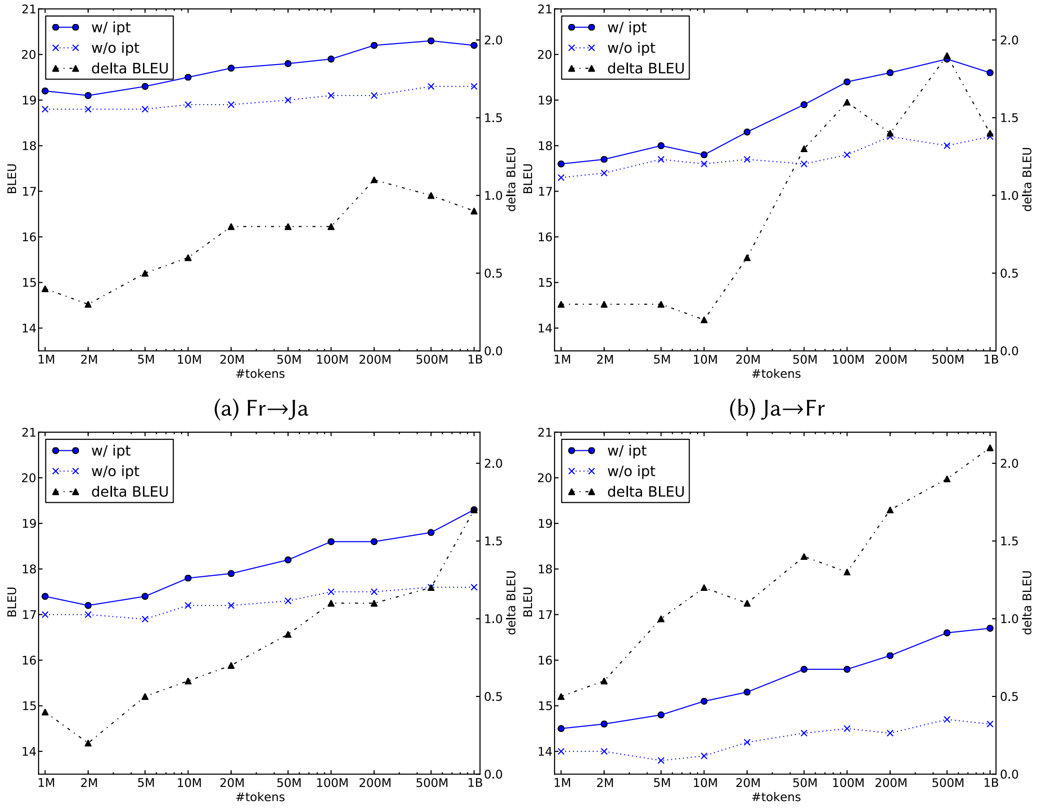
5.3 Effect of Monolingual Data Size

For many languages, large quantity of monolingual data, more than 100M tokens, for instance, are not available. In this section, we study the impact of the quantity of source and target monolingual data used to induce a phrase table. We expect that using a small amount of monolingual data will lead to the induction of a phrase table less useful for the decoder, as it will have a much reduced coverage and more poorly estimated features.

In this experiment, to assess the effect of the quantity of monolingual data, we fixed the quantity of parallel data used to train the bpt and to induce the ipt: the entire Tatoeba. For each translation direction, we induced a total of 10 ipts using different sizes of monolingual data randomly sampled from those used in the experiments in Section 4 (see Table 1): 1M, 2M, 5M, 10M, 20M, 50M, 100M, 200M, 500M, and 1B tokens. For each configuration, we used the same quantity of source and target monolingual data, except for the Es→Ja and Ja→Es ipts using 500M and 1B tokens. As our entire Spanish monolingual corpus contains only 446M tokens, we used all of it, while we used 500M and 1B tokens for Japanese.

We collected all the phrases with a lower frequency-based constraint, i.e., $K = 60$, than in our other experiments to collect at least 1k phrases even for the configurations with a very small quantity of monolingual data. Considering this experiment as a simulation of dealing with low-resource languages, we also built a language model used by the decoder only on the corresponding size of target monolingual data. For instance, in case we induce a phrase table with 1M tokens, the language model is trained on the same 1M tokens. Consequently, the baseline system using only the bpt will also be improved according to the increase of the size of the monolingual data used, because it will exploit a better estimated language model.

The results of our experiments are given in Figure 4. First, even with the smallest quantity of monolingual data, i.e., 1M tokens, to induce the ipt, we consistently obtain significant improvements over the baseline system for all translation directions. For this configuration, the largest improvement attains 0.5 BLEU points for the translation direction Ja→Es, while 0.3 BLEU points for the smallest improvement with the Ja→Fr translation direction. These results are surprising, because we have used an unusually small quantity of monolingual data to train the word embeddings used to compute the features presented in Section 3.2.1. Indeed, it is well-known that word embeddings are inaccurate when estimated on small corpora. Furthermore, with such a small size of target monolingual data, we collected far less target phrases to induce the phrase table. For instance, we collected only 1,677 phrases from 1M French data, while we obtained 393k phrases



(a) Fr→Ja

(b) Ja→Fr

(c) Es→Ja

(d) Ja→Es

Fig. 4. Results (BLEU) with regard to the size of the source and target monolingual corpus (number of tokens) used to induce our ipt. The curves denoted "delta BLEU" present the BLEU score improvements obtained by the configuration using the ipt over the system using only the bpt.

when using the entire French monolingual data (see Table 4). Among these 1,677 French phrases, 509 phrases were used by the decoder for the Fr→Ja translation direction, and 38 phrases were used to produce the output for the Ja→Fr translation direction.

We can deduce from the observed improvements that our ipt is able to propose more accurate translations than the bpt for frequent phrases, mainly relying on the features based on lexical translation probabilities to retrieve useful target phrases.

We also observed larger improvements over the baseline system when we used more monolingual data to induce a phrase table. This was expected as our features were better estimated and more diverse phrases were collected by introducing more monolingual data. Our experiments stopped at a size of 1B tokens, but even at that size we still observed greater improvements for the Es→Ja and Ja→Es translation directions compared to the configurations using 500M tokens. These results show that further improvements may be reachable if more monolingual data are available.

In summary, the level of improvement brought by our ipt is strongly related to the amount of monolingual data used. Moreover, our approach seems also useful even when only small quantities of source and target monolingual data are available. These results pave the way for a use of our approach to improve the translation quality also for truly low-resource languages.

Table 9. Results (BLEU) for Task A Using Both bpt and ipt

Source phrase collection	Fr→Ja	Ja→Fr	Es→Ja	Ja→Es
w/ the devtest	20.2 (12.0k)	19.6 (10.8k)	19.3 (10.8k)	16.7 (11.1k)
w/o the devtest	20.4 (9.6k)	**20.1** (9.1k)	19.5 (8.3k)	16.5 (9.4k)

Using the method presented in Section 3.1, we collected the source phrases from monolingual data including or not including the source side of the devtest. The numbers within parenthesis indicate the number of source phrases used for the induction of the ipt. Bold scores indicate statistical significance ($p < 0.01$) of the score over the system that has referred to the source side of the devtest to perform phrase collection.

5.4 Phrase Collection without Prior Knowledge of the Devtest

In the previous work on phrase table induction reviewed in Section 2, source phrases are directly extracted from the source side of the devtest sets, and we followed this convention in our experiments (see Section 4.3). However, this experimental setting is ideal in the sense that we know the devtest sets beforehand and can induce on-the-fly the phrase table according to the extracted source phrases, disregarding its computation time.

In this section, we report on an experiment to evaluate the impact of this setting by inducing a phrase table without referring to the source side of the devtest sets to collect the source phrases. To this end, we collected the set of phrases from only the external monolingual data in the corresponding language and retained only the phrases appearing in the source side of the devtest sets. A true ignorance of the devtest beforehand is out of our focus, because it requires inducing the phrase table considering the entire set of source phrases, i.e., without this prior filtering, while such a constraint only brings a drastic extension of the induction time. To avoid this unnecessary cost, we chose to filter the phrase set before the induction.

The experiments are conducted on task A using the entire Tatoeba corpus to train the bpt and all the monolingual data to induce the ipt. The results are presented in Table 9. The BLEU scores obtained without referring to the devtest sets to collect the source phrases are not significantly different from the results obtained by collecting source phrases referring to the devtest sets, except for the Ja→Fr translation direction. These results can be explained by the fact that we have used a large quantity of monolingual data that allow the extraction of a source phrase set with a relatively good coverage of the source side of the devtest. In contrast, when we included the source side of the devtest during the phrase collection, we collected more phrases, but those may appear rarely in the source monolingual data used to compute some of our features, such as word embeddings. In general, the embeddings of rare phrases are likely to be inaccurate. Moreover, rare source phrases containing rare tokens are even more likely to be infrequent, or absent, in the bilingual data used to estimate the lexical translation probabilities. As a consequence, we do not have any informative features for the phrase pairs involving such kind of source phrases. These source phrases are then going inevitably to be in the ipt paired only with incorrect target phrases.

These results point out that including the source side of the devtest during phrase collection is not necessary to induce an ipt useful for the decoder.

6 IN-DEPTH ANALYSIS OF THE USE OF AN IPT DURING DECODING

In our experiments above, we have retained in the ipt the 300-best target phrases for each source phrase (see Section 4.3). Although for some source phrases correct target phrases might be ranked higher by our classifier, as shown in Table 10, many of the retained phrase pairs in ipt were incorrect. This makes it challenging for the decoder to find the most appropriate one. By looking at some examples of translations generated by our systems, we observed that improvements were mainly

Table 10. Source Phrases and their 5-best Target Phrases Based on the Classifier's Score in the ipt for Task A, Fr→Ja, Using the Entire Tatoeba Corpus

source	langue française	une ballerine	cinquante-deux	la crème glacée	la jambe droite
target	フランス (France)	バレリーナ (ballerina)	五十二 (fifty two)	アイス (ice)	右手 (right hand)
	ドイツ (Germany)	ダンサー (dancer)	四十二 (forty two)	アイスクリーム (ice cream)	左足 (left leg)
	フランス語 (French)	美人 (beauty)	二十二 (twenty two)	クリーム (cream)	右足 (right leg)
	語 (word)	彼女 (she)	二十五 (twenty five)	ムース (mousse)	左 (left)
	ロシア (Russia)	女の子 (girl)	四十五 (forty five)	冷たい (cold)	左手 (left hand)

Underlined phrases are correct translations. English translations (in parentheses) are added for the sake of readability.

Table 11. Examples of Translations Generated on Task A, Ja→Fr and Ja→Es, Using the Entire Tatoeba Corpus, with or without the ipt

Ja→Fr	*source*	最初␣は␣私␣は␣何␣を␣し␣て␣よい␣か␣わから␣なかっ␣た␣。 (at first , I did not know what to do .)
	w/o ipt	au début , que j' ai .
	w/ ipt	au début , je ne sais pas quoi faire .
	reference	au début je ne savais pas quoi faire .
	source	ベジタリアン␣用␣の␣特別␣メニュー␣は␣あり␣ます␣か␣? (do you have a special menu for vegetarians ?)
	w/o ipt	le menu à ベジタリアン de vous ?
	w/ ipt	le menu végétarien il y a de spécial ?
	reference	avez-vous un menu spécial pour les végétariens ?
Ja→Es	*source*	彼女␣は␣地元␣の␣病院␣で␣看護␣婦␣として␣働い␣て␣いる␣。 (she works in a local hospital as a nurse .)
	w/o ipt	ella en el hospital de 地元 trabaja como enfermera .
	w/ ipt	ella está en el hospital de regional trabaja como enfermera .
	reference	ella trabaja como enfermera en el hospital local .
	source	彼女␣は␣年␣の␣わり␣に␣は␣おどろく␣ほど␣元気␣だ␣。 (she is surprisingly active considering her age .)
	w/o ipt	ella en わり del año es tan おどろく ánimo .
	w/ ipt	después de muchos años , ella es tan bien .
	reference	ella tiene una sorprendente vitalidad para su edad .

English translations for the Japanese source sentences are added (between parentheses) for readability.

due to the translations of originally OOV tokens. We also noticed improvements in translating sentences containing mostly frequent phrases. Our ipt was able to propose better translations than the bpt, even for frequent and already seen phrases in the parallel data used to train the bpt. Table 11 shows some examples of translations improved by using our ipt. To better understand how the decoder finds and uses good target phrases in the ipt, we performed an analysis of the

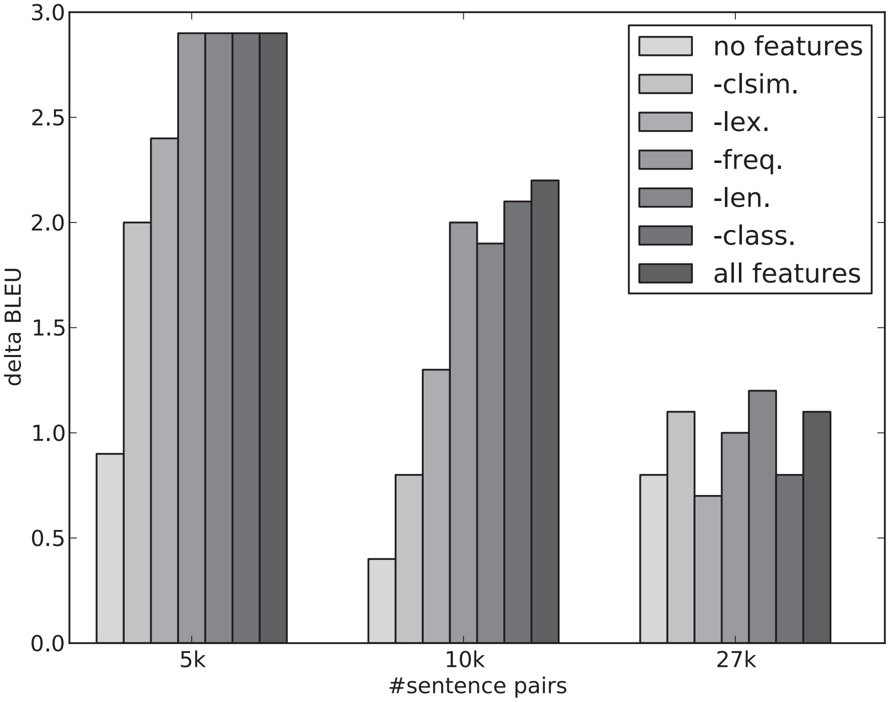


Fig. 5. Improvements (delta BLEU) obtained over the system with only the bpt on task A for the Fr→Ja translation direction, with regard to the number of parallel sentences used, after removing from the ipt different sets of features for decoding: those based on cross-lingual semantic similarity (clsim.), lexical translation probabilities (lex.), phrase frequency (freq.), phrase length (len.), and the classifier's score (class.). The configuration denoted "no features" means that an ipt is used but only associated with a global phrase penalty and contains no features for evaluating each phrase pair.

impact of its features. We also analyzed how often the ipt was used to generate the translation compared to the bpt.

First, we found out that the impact of the features of the ipt for decoding depends on the amount of parallel data used, as exemplified in Figure 5. The cross-lingual semantic similarity features were most important for decoding in the configurations using only 5k parallel sentences. For instance, on task A for the Fr→Ja translation direction, compared to the configuration using all the features, removing these features resulted in a drop of 0.9 BLEU points, while removing the features based on lexical translation probabilities led to a drop of 0.5 BLEU points. In contrast, for the system trained on all the Tatoeba sentence pairs, the features based on lexical translation probabilities were better estimated and then had a greater impact on decoding; removing them led to a drop of 0.4 BLEU points, while we observed no significant changes when removing cross-lingual semantic similarity features. Interestingly, even without any features in the ipt, we still observed an improvement: 0.9 BLEU points over the system with only the bpt for the configuration using 5k sentence pairs for instance. This highlights that the language model plays an important role in selecting appropriate phrase pairs in the ipt given some context.

As shown in Figure 6, our analysis of the generated translations revealed that the ipt is more often used by the systems trained on smaller amount of parallel data. For instance, on task A for Fr→Ja direction, using only 5k parallel sentences, 22.4% of the phrase pairs chosen by the decoder

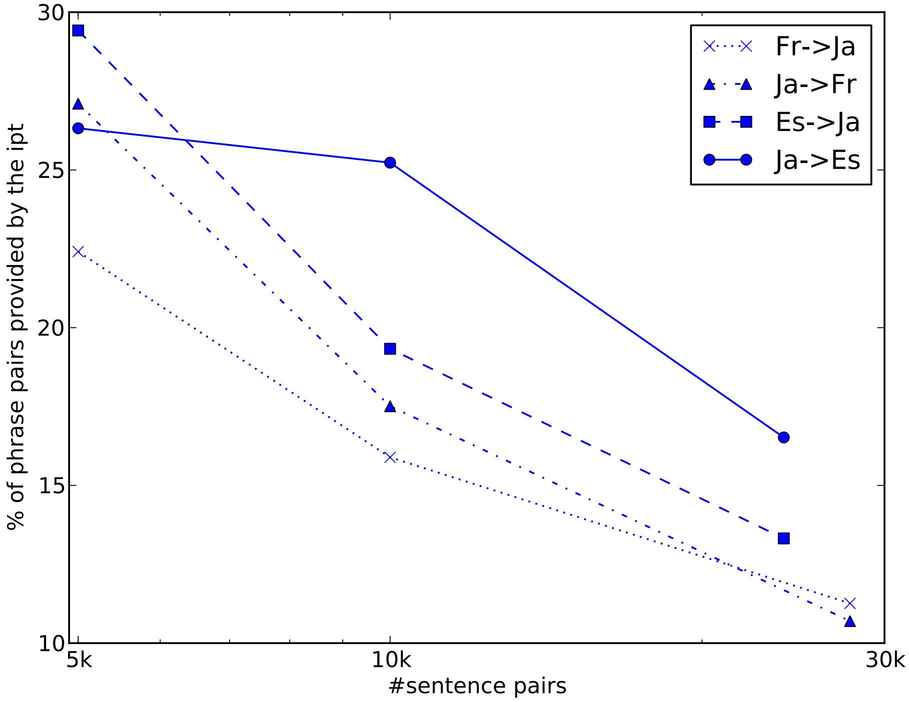


Fig. 6. Ratio of phrase pairs composing the translation generated by the decoder, provided by the `ipt`, with regard to the quantity of parallel sentences used to train the system.

were exclusively in the `ipt`, one third of them involved at least one phrase containing more than one token, while one fourth of them were covering a source phrase already seen in the `bpt`. The latter means that our approach can also propose new translations for source phrases already seen in the parallel data. When using the entire `Tatoeba` corpus, the decoder used our `ipt` less: only 11.3% of the phrase pairs used came from the `ipt` for Fr→Ja. In this configuration, the decoder used the `bpt` more, as it contained more phrase pairs associated with better estimated features.

We conclude from these observations that our `ipt` has more impact when only small amount of parallel data are available.

7 CONCLUSION AND FUTURE WORK

We showed that our phrase tables induced from completely unrelated monolingual data improved significantly the translation quality, especially in very low-resource conditions, using no comparable data but a very small amount of parallel data. To the best of our knowledge, our work is the first to successfully exploit phrase tables induced from monolingual data in such configurations. We also confirmed that a phrase table induced with our method can help an SMT system to generate translations of better quality, even when using a large amount of out-of-domain parallel data or a small amount of monolingual data to perform the induction.

As future work, we plan to evaluate our method on other language pairs to confirm its usefulness. We will also add new features to further improve translation quality. Moreover, we plan to study the induction of a reordering model for the phrase pairs in our induced phrase table. Such a model is usually helpful to produce a better translation quality. Furthermore, as parallel data may not be available for some specific domains, we will study the potential of our method to perform domain adaptation of an SMT system, using only monolingual data. Last but not least, we think that NMT may benefit from our work, especially in low-resource conditions where NMT performs poorly [25]. The phrase pairs proposed by our induced phrase tables may indeed help the NMT system to make better translation choice. This could be realized with an hybrid search fashion, such as the one proposed by Dahlmann et al. [7].

ACKNOWLEDGMENTS

We thank all the reviewers and the Associate Editor for useful comments and suggestions.

REFERENCES

[1] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the International Conference on Computational Linguistics (COLING'16)*.

[2] Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'12)*.

[3] Chenhui Chu and Sadao Kurohashi. 2016. Paraphrasing out-of-vocabulary words with word embeddings and semantic lexicons for low resource statistical machine translation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'16)*.

[4] Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the Annual Conference of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'11)*.

[5] Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'07)*.

[6] Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. 2015. Trans-gram, fast crosslingual word-embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*.

[7] Leonard Dahlmann, Evgeny Matusov, Pavel Petrushkov, and Shahram Khadivi. 2017. Neural machine translation leveraging phrase-based models in a hybrid search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

[8] Hal Daumé III and Jagadeesh Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the Annual Conference of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'11)*.

[9] Meiping Dong, Yang Liu, Huanbo Luan, Maosong Sun, Tatsuya Izuha, and Dakun Zhang. 2015. Iterative learning of parallel lexicons and phrases from non-parallel corpora. In *Proceedings of the International Joint Conferences on Artificial Intelligence Organization (IJCAI'15)*.

[10] Qing Dou and Kevin Knight. 2012. Large scale decipherment for out-of-domain machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the Conference on Natural Language Learning (EMNLP-CoNLL'12)*.

[11] Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2016. Learning crosslingual word embeddings without bilingual corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*.

[12] Pascale Fung. 1995. Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. In *Proceedings of the 3rd Workshop on Very Large Corpora*.

[13] Ulrich Germann. 2001. Building a statistical machine translation system from scratch: How much bang for the buck can we expect? In *Proceedings of the Conference of the Association for Computational Linguistics Workshop on Data-Driven Methods in Machine Translation*.

[14] A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of the Conference of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'08)*.

[15] Jingyi Han and Núria Bel. 2016. Towards producing bilingual lexica from monolingual corpora. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'16)*.

[16] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified kneserney language model estimation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'13)*.

[17] Sanjika Hewavitharana and Stephan Vogel. 2016. Extracting parallel phrases from comparable data for machine translation. *Nat. Lang. Eng.* 22, 4 (2016), 549–573.

[18] Ann Irvine and Chris Callison-Burch. 2013. Supervised bilingual lexicon induction with multiple monolingual signals. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'13)*.

[19] Ann Irvine and Chris Callison-Burch. 2014. Hallucinating phrase translations for low resource MT. In *Proceedings of the Conference on Natural Language Learning (CoNLL'14)*.

[20] Ann Irvine and Chris Callison-Burch. 2016. End-to-end statistical machine translation with zero or small parallel texts. *Nat. Lang. Eng.* 22, 4 (2016), 517–548.

[21] Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and the Conference on Natural Language Learning (EMNLP-CoNLL'07)*.

[22] Alex Klementiev, Ann Irvine, Chris Callison-Burch, and David Yarowsky. 2012. Toward statistical machine translation without parallel corpora. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL'12)*.

[23] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'07)*.

[24] Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the Conference of the Association for Computational Linguistics Workshop on Unsupervised Lexical Acquisition*.

[25] Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the 1st Workshop on Neural Machine Translation*.

[26] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR* abs/1309.4168 (2013).

[27] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS'13)*.

[28] Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science* 34, 8 (2010), 1388–1429.

[29] Malte Nuhn, Arne Mauser, and Hermann Ney. 2012. Deciphering foreign language by combining language models and context vectors. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'12)*.

[30] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'02)*.

[31] Peyman Passban, Qun Liu, and Andy Way. 2016. Enriching phrase tables for statistical machine translation using mixed embeddings. In *Proceedings of the International Conference on Computational Linguistics (COLING'16)*.

[32] Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'95)*.

[33] Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the Annual Conference of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'11)*.

[34] Avneesh Saluja, Hany Hassan, Kristina Toutanova, and Chris Quirk. 2014. Graph-based semi-supervised learning of translation models from monolingual data. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'14)*.

[35] Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'13)*.

[36] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*.

[37] Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'16)*.

[38] Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'07)*.

[39] Ivan Vulić and Anna Korhonen. 2016. On the role of seed lexicons in learning bilingual word embeddings. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'16)*.

[40] Ivan Vulić and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL'15)*.

[41] Ivan Vulić and Marie-Francine Moens. 2016. Bilingual distributed word representations from document-aligned comparable data. *Journal of Articial Intelligence Research* 55 (2016), 953–994.

[42] Kai Zhao, Hany Hassan, and Michael Auli. 2015. Learning translation models from monolingual continuous representations. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'15).*

Received June 2017; revised September 2017; accepted November 2017