Improving Statistical Machine Translation with a Multilingual Paraphrase Database

Ramtin Mehdizadeh Seraj, Maryam Siahbani, Anoop Sarkar
School of Computing Science
Simon Fraser University
Burnaby BC. Canada
rmehdiza,msiahban,anoop@cs.sfu.ca

Abstract

The multilingual Paraphrase Database (PPDB) is a freely available automatically created resource of paraphrases in multiple languages. In statistical machine translation, paraphrases can be used to provide translation for out-of-vocabulary (OOV) phrases. In this paper, we show that a graph propagation approach that uses PPDB paraphrases can be used to improve overall translation quality. We provide an extensive comparison with previous work and show that our PPDB-based method improves the BLEU score by up to 1.79 percent points. We show that our approach improves on the state of the art in three different settings: when faced with limited amount of parallel training data; a domain shift between training and test data; and handling a morphologically complex source language. Our PPDB-based method outperforms the use of distributional profiles from monolingual source data.

1 Introduction

Translation coverage is a major concern in statistical machine translation (SMT) which relies on large amounts of parallel, sentence-aligned text. In (Callison-Burch et al., 2006), even with a training data size of 10 million word tokens, source vocabulary coverage in unseen data does not go above 90%. The problem is worse with multi-word OOV phrases. Copying OOVs to the output is the most common solution. However, even noisy translations of OOVs can improve reordering and language model scores (Zhang et al., 2012). Transliteration is useful but not a panacea for the OOV problem (Irvine and Callison-Burch, 2014b). We find and remove the named entities, dates, etc. in the source and focus on the use of paraphrases to help translate the remaining OOVs. In Sec. 5.2 we show that handling such OOVs correctly does improve translation scores.

In this paper, we build on the following research: *Bilingual lexicon induction* is the task of learning translations of words from monolingual data in source and target languages (Schafer and Yarowsky, 2002; Koehn and Knight, 2002; Haghighi et al., 2008). The *distributional profile* (DP) approach uses context vectors to link words as potential paraphrases to translation candidates (Rapp, 1995; Koehn and Knight, 2002; Haghighi et al., 2008; Garera et al., 2009). DPs have been used in SMT to assign translation candidates to OOVs (Marton et al., 2009; Daumé and Jagarlamudi, 2011; Irvine et al., 2013; Irvine and Callison-Burch, 2014a). Graph-based semi-supervised methods extend this approach and propagate translation candidates across a graph with phrasal nodes connected via weighted paraphrase relationships (Razmara et al., 2013; Saluja et al., 2014; Zhao et al., 2015). Saluja et al. (2014) extend paraphrases for SMT from the words to phrases, which we also do in this work. *Bilingual pivoting* uses parallel data instead of context vectors for paraphrase extraction (Mann and Yarowsky, 2001; Schafer and Yarowsky, 2002; Bannard and Callison-Burch, 2005; Callison-Burch et al., 2006; Zhao et al., 2008; Callison-Burch, 2008). Ganitkevitch and Callison-Burch (2014) published a large-scale multilingual Paraphrase Database (PPDB) http://paraphrase.org which includes lexical, phrasal, and syntactic paraphrases (available for 22 languages with up to 170 million paraphrases each).

To our knowledge, this paper is the first comprehensive study of the use of PPDB for statistical machine translation model training. Our framework has three stages: 1) a novel graph construction approach for PPDB paraphrases linked

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1379–1390,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

with phrases from parallel training data. 2) Graph propagation that uses PPDB paraphrases. 3) An SMT model that incorporates new translation candidates. Sec. 3 explains these three stages in detail.

Using PPDB has several advantages: 1) Resources such as PPDB can be built and used for many different tasks including but not limited to SMT. 2) PPDB contains many features that are useful to rank the strength of a paraphrase connection and with more information than distributional profiles. 3) Paraphrases in PPDB are often better than paraphrases extracted from monolingual or comparable corpora because a large-scale multilingual paraphrase database such as PPDB can pivot through a large amount of data in many different languages. It is not limited to using the source language data for finding paraphrases which distinguishes it from previous uses of paraphrases for SMT.

PPDB is a natural resource for paraphrases. However, PPDB was not built with the specific application to SMT in mind. Other applications such as text-to-text generation have used PPDB (Ganitkevitch et al., 2011) but SMT brings along a specific set of concerns when using paraphrases: translation candidates should be transferred suitably across paraphrases. There are many cases, e.g. when faced with different word senses where transfer of a translation is not appropriate. Our proposed methods of using PPDB use graph propagation to transfer translation candidates in a way that is sensitive to SMT concerns.

In our experiments (Sec. 5) we compare our approach with the state-of-the-art in three different settings in SMT: 1) when faced with limited amount of parallel training data; 2) a domain shift between training and test data; and 3) handling a morphologically complex source language. In each case, we show that our PPDB-based approach outperforms the distributional profile approach.

2 Paraphrase Extraction

Our goal is to produce translations for OOV phrases by exploiting paraphrases from the multilingual PPDB (Ganitkevitch and Callison-Burch, 2014) by using graph propagation. Since our approach relies on phrase-level paraphrases we compare with the current state of the art approaches that use monolingual data and distributional profiles to construct paraphrases and use graph propagation (Razmara et al., 2013; Saluja et al., 2014).

2.1 Paraphrases from Distributional Profiles

A *distributional profile* (DP) of a word or phrase was first proposed in (Rapp, 1995) for SMT. Given a word f, its distributional profile is:

$$DP(f) = \{\langle A(f, w_i)\rangle \mid w_i \in V\}$$

V is the vocabulary and the surrounding words w_i are taken from a monolingual corpus using a fixed window size. We use a window size of 4 words based on the experiments in (Razmara et al., 2013). DPs need an association measure $A(\cdot, \cdot)$ to compute distances between potential paraphrases. A comparison of different association measures appears in (Marton et al., 2009; Razmara et al., 2013; Saluja et al., 2014) and our preliminary experiments validated the choice of the same association measure as in these papers, namely *Pointwise Mutual Information* (Lin, 1998) (PMI). For each potential context word w_i:

$$A(f, w_i) = log_2 \frac{P(f, w_i)}{P(f)P(w_i)} \quad (1)$$

To evaluate the similarity between two phrases we use cosine similarity. The cosine coefficient of two phrases f_1 and f_2 is:

$$S(f_1, f_2) = cos(DP(f_1), DP(f_2)) = \frac{\sum_{w_i \in V} A(f_1, w_i)A(f_2, w_i)}{\sqrt{\sum_{w_i \in V} A(f_1, w_i)^2}\sqrt{\sum_{w_i \in V} A(f_2, w_i)^2}} \quad (2)$$

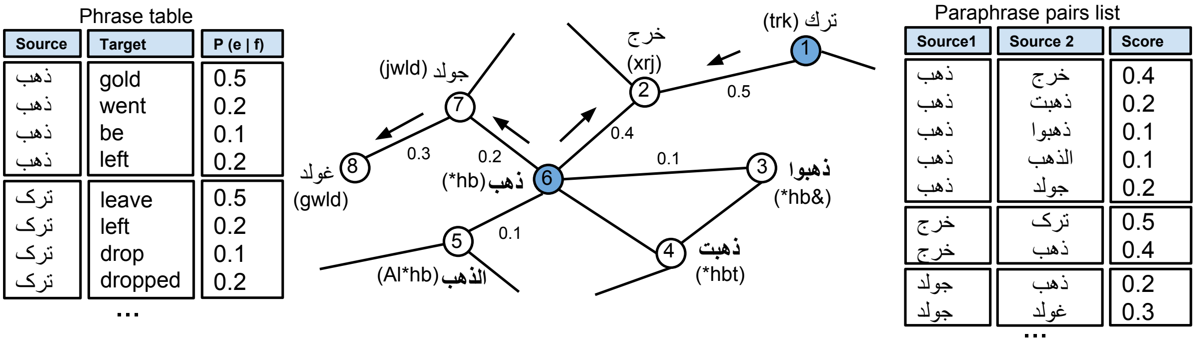
where V is the vocabulary. Note that in Eqn. (2) w_i's are the words that appear in the context of f_1 or f_2, otherwise the PMI values would be zero.

Considering all possible candidate paraphrases is very expensive. Thus, we use the heuristic applied in previous works (Marton et al., 2009; Razmara et al., 2013; Saluja et al., 2014) to reduce the search space. For each phrase we keep candidate paraphrases which appear in one of the surrounding context (e.g. *Left__Right*) among all occurrences of the phrase.

2.2 Paraphrases from bilingual pivoting

Bilingual pivoting uses parallel corpora between the source language, F, and a pivot language T. If two phrases, f_1 and f_2, in a same language are paraphrases, then they share a translation in other languages with $p(f_1|f_2)$ as a paraphrase score:

$$S(f_1, f_2) = p(f_1|f_2) = \sum_t p(f_1|t)p(t|f_2) \quad (3)$$



Phrase table

Source	Target	P (e | f)
ذهب	gold	0.5
ذهب	went	0.2
ذهب	be	0.1
ذهب	left	0.2
ترک	leave	0.5
ترک	left	0.2
ترک	drop	0.1
ترک	dropped	0.2
	...	

Paraphrase pairs list

Source1	Source 2	Score
ذهب	خرج	0.4
ذهب	ذهبت	0.2
ذهب	ذهبوا	0.1
ذهب	الذهب	0.1
ذهب	جولد	0.2
خرج	ترک	0.5
خرج	ذهب	0.4
جولد	ذهب	0.2
جولد	غولد	0.3
	...	

Figure 2: A small sample of the real graph constructed from the Arabic PPDB for Arabic to English translation. Filled nodes (1 and 6) are phrases from the SMT phrase table (unfilled nodes are not). Edge weights are set using a log-linear combination of scores from PPDB. Phrase #6 has different senses (‘gold’ or ‘left’); and it has a paraphrase in phrase #7 for the ‘gold’ sense and a paraphrase in phrase #2 for the ‘left’ sense. After propagation, phrase #2 receives translation candidates from phrase #6 and phrase #1 reducing the probability of translation from unrelated senses (like the ‘gold’ sense). Phrase #8 is a misspelling of phrase #7 and is also captured as a paraphrase. Phrase #6 propagates translation candidates to phrase #8 through phrase #7. Morphological variants of phrase #6 (shown in bold) also receive translation candidates through graph propagation giving translation candidates for morphologically rich OOVs.

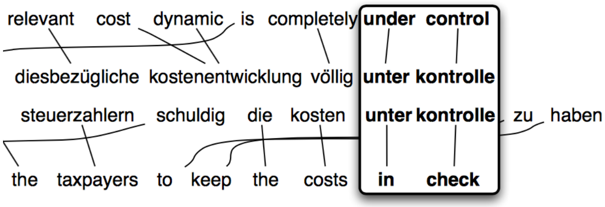


Figure 1: English paraphrases extracted by pivoting over German shared translation (Bannard and Callison-Burch, 2005).

where t is a phrase in language T. $p(f_1|t)$ and $p(t|f_2)$ are taken from the phrase table extracted from parallel data for languages F and T. In Fig. 1 from (Bannard and Callison-Burch, 2005) we see that paraphrase pairs like (*in check*, *under control*) can be extracted by pivoting over the German phrase *unter kontrolle*.

The multilingual Paraphrase Database (PPDB) (Ganitkevitch and Callison-Burch, 2014) is a published resource for paraphrases extracted using bilingual pivoting. It leverages syntactic information and other resources to filters and scores each paraphrase pair using a large set of features. These features can be used by a log linear model to score paraphrases (Zhao et al., 2008). We used a linear combination of these features using the equation in Sec. 3 of (Ganitkevitch and Callison-Burch, 2014) to score paraphrase pairs. PPDB version 1 is broken into different levels of coverage. The smaller sizes contain only better-scoring, high-precision paraphrases, while larger sizes aim for high coverage.

Algorithm 1 PPDB Graph Propagation for SMT

```
PhrTable = PhraseTableGeneration();
ParaDB = ParaphraseExtraction();                       (Sec. 2)
InitGraph = GraphConstruct(PhrTable, ParaDB); (Sec. 3.1)
PropGraph = GraphPropagation(InitGraph);         (Sec. 3.2)
for phrase $\in$ {OOVs} do
    newTrans = TranslationFinder(PropGraph, phrase);
    Augment(PhrTable, newTrans);                       (Sec. 3.3)
TuneMT(PhrTable);
```

3 Methodology

After paraphrase extraction we have paraphrase pairs, (f_1, f_2) and a score $S(f_1, f_2)$ we can induce new translation rules for OOV phrases using the steps in Algo. (1): 1) A graph of source phrases is constructed as in (Razmara et al., 2013); 2) translations are propagated as labels through the graph as explained in Fig. 2; and 3) new translation rules obtained from graph-propagation are integrated with the original phrase table.

3.1 Graph Construction

We construct a graph $G(V, E, W)$ over all source phrases in the paraphrase database and the source language phrases from the SMT phrase table extracted from the available parallel data. V corresponds to the set of vertices (source phrases), E is the set of edges between phrases and W is weight of each using the score function S defined in Sec. 2. V has two types of nodes: seed (labeled) nodes, V_s, from the SMT phrase table, and regular nodes, V_r. Note that in this step OOVs are part of these regular nodes, and we try to find translation in the propagation step for all of these regular nodes. In graph construction and propagation, we do not know which phrasal nodes correspond to OOVs in the dev and test set. Fig. 2 shows a small slice of the actual graph used in one of our experiments; This graph is constructed using the paraphrase database on the right side of the figure. Filled nodes have a distribution over translations (the possible "labels" for that node). In our setting, we consider the translation e to be the "label" and so we propagate the labeling distribution $p(e|f)$ which is taken from the feature function for the SMT log-linear model that is taken from the SMT phrase table and we propagate this distribution to unlabeled nodes in the graph.

3.2 Graph Propagation

Considering the translation candidates of known phrases in the SMT phrase table as the "labels" we apply a soft label propagation algorithm in order to assign translation candidates to "unlabeled" nodes in the graph, which include our OOV phrases. As described by the example in Fig. 2 we wish two outcomes: 1) transfer of translations (or "labels") to unlabeled nodes (OOV phrases) from labeled nodes, and 2) smoothing the label distribution at each node. We use the Modified Adsorption (MAD) algorithm (Talukdar and Crammer, 2009) for graph propagation. Suppose we have m different possible labels plus one *dummy label*, a soft label $\hat{Y} \in \Delta^{m+1}$ is a $m+1$ dimension probability vector. The dummy label is used when there is low confidence on correct labels. Based on MAD, we want to find soft label vectors for each node by optimizing the objective function below:

$$\begin{aligned}\min_{\hat{Y}} \mu_1 \sum_{v \in V_s} P_{1,v} ||Y_v - \hat{Y}||_2^2 + \\ \mu_2 \sum_{v \in V, u \in N(v)} P_{2,v} W_{v,u} ||\hat{Y_v} - \hat{Y_u}||_2^2 + \\ \mu_3 \sum_{v \in V} P_{3,v} ||\hat{Y_v} - R_v||_2^2\end{aligned} \quad (4)$$

In this objective function, μ_i and $P_{i,v}$ are hyperparameters ($\forall v : \Sigma_i P_{i,v} = 1$). $R_v \in \Delta^{m+1}$ is our prior belief about labeling. First component of the function tries to minimize the difference of new distribution to the original distribution for the seed nodes. The second component insures that nearby neighbours have similar distributions, and the final component is to make sure that the distribution does not stray from a prior distribution. At the end of propagation, we wish to find a label distribution for our OOV phrases. We describe in Sec. 4.2.2 the reasons for choosing MAD over other graph propagation algorithms. The MAD graph propagation generalizes the approach used in (Razmara et al., 2013). The Structured Label Propagation algorithm (SLP) was used in (Saluja et al., 2014; Zhao et al., 2015) which uses a graph structure on the target side phrases as well. However, we have found that in our diverse experimental settings (see Sec. 5) MAD had two properties we needed compared to SLP: one was the use of graph random walks which allowed us to control translation candidates and MAD also has the ability to penalize nodes with a large number of edges (also see Sec. 4.2.2).

3.3 Phrase Table Integration

After propagation, for each potential OOV phrase we have a list of possible translations with corresponding probabilities. A potential OOV is any phrase which does not appear in training, but could appear in unseen data. We do not look at the dev or test data to produce the augmented phrase table. The original phrase table is now augmented with new entries providing translation candidates for potential OOVs; Last column in Table 2 shows how many entries have been added to the phrase table for each experimental settings. A new feature is added to the standard SMT log-linear discriminative model and introduced into the phrase table. This new feature is set to either 1.0 for the phrase table entries that already existed; or ℓ_i which is the log probability (from graph propagation) for the translation candidate i for potential OOVs. In case the dummy label exists with high probability or the label distribution is uniform, an identity rule is added to the phrase table (copy over source to target).

4 Analysis of the Framework

4.1 Propagation of poor translations

Automatic paraphrase extraction generates many possible paraphrase candidates and many of them are likely to be false positives for finding translation candidates for OOVs. Distributional profiles rely on context information which is not sufficient to derive accurate paraphrases for many phrases and this results in many low quality paraphrase candidates. Bilingual pivoting uses word alignments which can also introduce errors depending on the size and quality of the bilingual data used. Alignment errors also introduce poor translations.

Size	Nodes	Edges	Max Neigh.	Ave Neigh.
S	23K	31K	32	1.38
M	41K	69K	33	1.69
L	74K	199K	67	2.69
XL	103K	548K	330	5.33
XXL	122K	2073K	1231	16.968
XXXL	125K	7558K	5255	60.27

Table 1: Statistics of the graph constructed using the English lexical PPDB. We have built similar graphs for French and Arabic.

In graph propagation, these errors may be propagated and result in poor translations for OOVs.

We could address this issue by aggressively pruning the potential paraphrase candidates to improve the precision. However, this results in a dramatic drop in coverage and many OOV phrases do not obtain any translation candidates. We use a combination of the following three steps to augment our graph propagation framework.

4.1.1 Graph pruning and PPDB sizes

Pruning the graph avoids error propagation by removing unreliable edges. Pruning removes edges with an edge weight lower than a minimum threshold or by limiting the number of neighbours to the top-K edges (Talukdar, 2009). PPDB has different sizes with different levels of accuracy and coverage. We can do graph pruning simply by choosing to use different sizes of PPDB. As we can see in Fig. 3 results vary from language to language depending on the pruning used. For instance, the L size results in the best score for French-English. We choose the best size of PPDB for each language based on a separate held-out set and independently from each of the SMT-based tasks in our experimental results. Our conclusion from our experiments with the different sizes of PPDB is that removing phrases (or nodes in our graph) is not desirable. However, removing unreliable edges is useful. As seen in Table 1, increasing the size of PPDB leads to a rapid increase in nodes followed by a larger number of edges in the very large PPDB sizes.

4.1.2 Pruning the translation candidates

Another solution to the error propagation issue is to propagate all translation candidates but when providing translations to OOVs in the final phrase table to eliminate all but the top L translations for each phrase (which is the usual *ttable* limit in phrase-based SMT (Koehn et al., 2003)). Based on a development set, separate from the test sets we used, we found that the best value of L was 10.

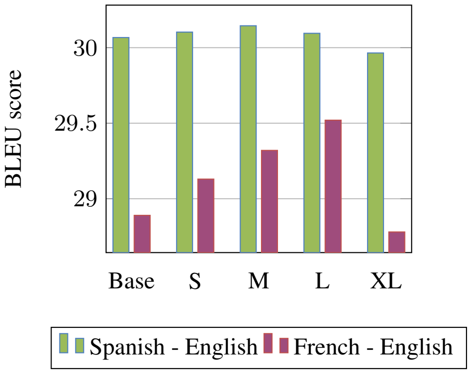


Figure 3: Effect of PPDB size on improving BLEU score for Spanish and French

4.1.3 External Resources for Filtering

Applying more informative filters can be also used to improve paraphrase quality. This can be done through additional features for paraphrase pairs. For example, edit distance can be used to capture misspelled paraphrases. We use a Named Entity Recognizer to exclude names, numbers and dates from the paraphrase candidates. Even after removing these tokens, 3.32% of tokens of test set are still OOVs . In addition, we use a list of stop words to remove nodes which have too many connections. These two filters improve our results (more in Sec. 5).

4.2 Path sensitivity

Graph propagation has been used in many NLP tasks like POS tagging, parsing, etc. but propagating translations in a graph as labels is much more challenging. Due to huge number of possible labels (translations) and many low quality edges, it is very likely that many wrong translations are rapidly propagated in few steps. Razmara et al. (2013) show that unlabeled nodes inside the graph, called *bridge nodes*, are useful for the transfer of translations when there is no other connection between an OOV phrase and a node with known translation candidates. However, they show that using the full graph with long paths of bridge nodes hurts performance. Thus the propagation has to be constrained using *path sensitivity*. Fig. 4 shows this issue in a part of an English para-

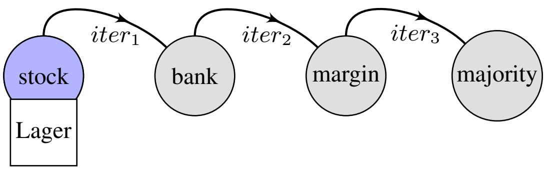


Figure 4: Sensitivity issue in graph propagation for translations. "Lager" is a translation candidate for "stock", which is transferred to "majority" after 3 iterations.

phrase graph. After three iterations, German translation "Lager" reaches "majority" which is totally irrelevant as a translation candidate. Transfer of translation candidates should prefer close neighbours and only with a very low probability to other nodes in the graph.

4.2.1 Pre-structuring the graph

Razmara et al. (2013) avoid a fully connected graph structure. They pre-structure the graph into bipartite graphs (only connections between phrases with known translation and OOV phrases) and tripartite graphs (connections can also go from a known phrasal node to an OOV phrasal node through one node that is a paraphrase of both but does not have translations, i.e. it is an unlabeled node). In these pre-structured graphs there are no connections between nodes of the same type (known, OOV or unlabeled). We apply this method in our low resource setting experiments (Sec. 5.3) to compare our bipartite and tripartite results to Razmara et al. (2013). In the rest of the experiments we use the tripartite approach since it outperforms the bipartite approach.

4.2.2 Graph random walks

Our goal is to limit the number of hops in the propagation of translation candidates preferring closely connected and highly probable edge weights. Optimization for the Modified Adsorption (MAD) objective function in Sec. 3.2 can be viewed as a controlled random walk (Talukdar et al., 2008; Talukdar and Crammer, 2009). This is formalized as three actions: *inject*, *continue* and *abandon* with corresponding pre-defined probabilities P_{inj}, P_{cont} and P_{abnd} respectively as in (Talukdar and Crammer, 2009). A random walk through the graph will transfer labels from one node to another node, and probabilities P_{cont} and P_{abnd} control exploration of the graph. By reducing the values of P_{cont} and increasing P_{abnd} we can control the label propagation process to optimize the quality of translations for OOV phrases. Again, this is done on a held-out development set and not on the test data. The optimal values in our experiments for these probabilities are $P_{inj} = 0.9, P_{cont} = 0.001, P_{abnd} = 0.01$.

4.2.3 Early stopping of propagation

In Modified Adsorption (MAD) (see Sec. 3.2) nodes in the graph that are closely linked will tend to similar label distributions as the number of iterations increase (even when the path lengths increase). In our setting, smoothing the label distribution helps in the first few iterations, but is harmful as the number of iterations increase due to the factors shown in Fig. 4. We use *early stopping* which limits the number of iterations. We varied the number of iterations from 1 to 10 on a held-out dev set and found that 5 iterations was optimal.

5 Evaluation

We first show the effect of OOVs on translation quality, then evaluate our approach in three different SMT settings: low resource SMT, domain shift, and morphologically complex languages. In each case, we compare results of using paraphrases extracted by Distributional Profile (DP) and PPDB in an end-to-end SMT system.

Important: no subset of the test data sentences are used in the bilingual corpora for paraphrase extraction process.

5.1 Experimental Setup

We use CDEC[1] (Dyer et al., 2010) as an end-to-end SMT pipeline with its standard features[2]. `fast_align` (Dyer et al., 2013) is used for word alignment, and weights are tuned by minimizing BLEU loss on the dev set using MIRA (Crammer and Singer, 2003). This setup is used for most of our experiments: oracle (Sec. 5.2), domain adaptation (Sec. 5.4) and morphologically complex languages (Sec. 5.5). But as we wish to fairly compare our approach with Razmara et al. (2013) on low resource setting, we follow their setup in Sec. 5.3: Moses (Koehn et al., 2007) as SMT pipeline, GIZA++ (Och and Ney, 2003) for word alignment and MERT (Och, 2003) for tuning. We add our own feature to the SMT log-linear model as described in Sec. 3.3.

[1] http://www.cdec-decoder.org

[2] EgivenFCoherent, SampleCountF, CountEF, MaxLexFgivenE, MaxLexEgivenF, IsSingletonF, IsSingletonEF

Experiments	OOV type/token	Rules added
Case 1	1830 / 2163	7.0K
Case 2 - Med.	2294 / 4190	7.8K
Case 2 - Sci.	5272 / 14121	10.4K
Case 3	1543 / 1895	8.1K

Table 2: Statistics of settings in Sec. 5. Last column shows how many rules added in the phrase table integration step.

KenLM (Heafield, 2011) is used to train a 5-gram language model on English Gigaword (V5: LDC2011T07). For scalable graph propagation we use the Junto framework[3]. We use maximum phrase length 10. For our experiments we use the Hadoop distributed computing framework executed on a cluster with 12 nodes (each node has 8 cores and 16GB of RAM). Each graph propagation iteration takes about 3 minutes.

For French, we apply a simple heuristic to detect named entities: words that are capitalized in the original dev/test set that do not appear at the beginning of a sentence are named entities. Based on eyeballing the results, this works very well in our data. For Arabic, AQMAR is used to exclude named-entities (Mohit et al., 2012). For each of the experimental settings below we show the OOV statistics in Table 2.

5.2 Impact of OOVs: Oracle experiment

This oracle experiment shows that translation of OOVs beyond named entities, dates, etc. is potentially very useful in improving output translation. We trained a SMT system on 10K French-English sentences from the Europarl corpus(v7) (Koehn, 2005). WMT 2011 and WMT 2012 are used as dev and test data respectively. Table 4 shows the results in terms of BLEU on dev and test. The first row is baseline which simply copies OOVs to output. The second and third rows show the result of augmenting phrase-table by adding translations for single-word OOVs and phrases containing OOVs. The last row shows the oracle result where dev and test sentences exist inside the training data and all the OOVs are known (Fully observers cannot avoid model and search errors).

5.3 Case 1: Limited Parallel Data

In this experiment we use a setup similar to (Razmara et al., 2013). To have fair comparison, we use 10K French-English parallel sentences, randomly chosen from Europarl to train translation system, as reported in (Razmara et al., 2013). ACL/WMT 2005[4] is used for dev and test data. We re-implement their paraphrase extraction method (DP) to extract paraphrases from French side of Europarl (2M sentences). We use unigram nodes to construct graphs for both DP and PPDB. In bipartite graphs, each node is connected to at most 20 nodes. For tripartite graphs, each node is connected to 15 labeled and 5 unlabeled nodes.

Fr-En	Dev	Test
Baseline	27.90	28.08
+ Lexical OOV	28.10	28.31
+ Phrasal OOV	28.50	28.85
Fully observed	46.88	49.21

Table 4: The impact of translating OOVs.

For intrinsic evaluation, we use Mean-Reciprocal-Rank (MRR) and Recall. MRR is the mean of reciprocal rank of the candidate list compared to the gold list (Eqn. 5). Recall shows percentage of gold list covered by the candidate list (Eqn. 6). Gold translations for OOVs are given by concatenating the test data to training and running a word aligner.

$$\text{MRR} = \frac{1}{|O|} \sum_{i=1}^{|O|} \frac{1}{rank_i} \text{ for } O = \{\text{OOVs}\} \quad (5)$$

$$\text{Recall} = \frac{|\{\text{gold list}\} \cap \{\text{candidate list}\}|}{|\{\text{gold list}\}|} \quad (6)$$

Table 5 compares DP and PPDB in terms of BLEU, MRR and Recall. It indicates that PPDB (large size) outperforms DP in both intrinsic and extrinsic evaluation measures. Although tripartite graph did not improve the results for DP, it results in statistically significantly better BLEU score for PPDB in comparison to DP (evaluated by MultEval (Clark et al., 2011)). Thus we use tripartite graph in the rest of experiments. The last row in the table shows the result of combining DP and PPDB by multiplying the normalized scores of both paraphrase lists.

This setting is included for three reasons: 1) we exploit the small data size to explore different choices in our approach such as, e.g. choosing bipartite versus tripartite graph structures; 2)

[3] Junto : https://github.com/parthatalukdar/junto

[4] http://www.statmt.org/wpt05/mt-shared-task/

OOV	PPDB NNs	DP NNs	Reference sentence	PPDB output	DP output
procédés	processus	méthodes outils matériaux	... an agreement on **procedures** in itself is a good thing ...	... an agreement on **the procedure** is a good ...	... an agreement on **products** is a good ...
quantique	quantiques	-	... allowed us to achieve **quantum** degeneracy ...	... allowed **quantum** degeneracy ...	... **quantique** allowed degeneracy ...
mlzm	mlzmA	ADTr	... voted 97-0 last week for a non-**binding** resolution ...	... voted 97 last week on **not binding** resolution ...	... voted 97 last week on **having** resolution ...

Table 3: Examples comparing DP versus PPDB outputs on the test sets. NNs refer to nearest neighbours in the graph for OOV phrase. Each row respectively corresponds to experimental settings (cases 1 to 3).

System	MRR	Recall	BLEU
baseline	-	-	28.89
DP-bipartite	5.34	11.90	29.27
DP-tripartite	5.34	11.95	29.27
PPDB$_{\text{fr}}$ (L)-bipartite	**12.05**	22.08	29.46
PPDB$_{\text{fr}}$ (L)-tripartite	10.22	**22.87**	**29.52**
Combined-tripartite	-	-	29.28

Table 5: Results of PPDB and DP techniques.

to show how well our PPDB approach does compared to the DP approach in terms of MRR and recall; and 3) to show applicability of our approach for a low-resource language. However we used French instead of a language which is truly resource-poor due to the lack of available paraphrases for a true resource poor language, e.g. Malagasy.

5.4 Case 2: Domain Adaptation

Domain adaptation is another case that suffers from massive number of OOVs. We compare our approach with Marginal Matching (Irvine et al., 2013), a state of the art approach in SMT domain adaptation. We use their setup and data and compare our results to their reported results (Irvine et al., 2013). 250K lines of Hansard parliamentary proceeding are used for training MT. Dev and test sets are available for two different domains: Medical and Science domains. For medical domain random subset of EMEA corpus (Tiedemann, 2009) and for the science domain a corpus of scientific articles (Carpuat et al., 2012) has been used. Unigram paraphrases using DP are extracted from French side of Europarl.

Table 6 compares the results in terms of BLEU score. In both medical and science domains, graph-propagation approach using PPDB (large) performs significantly better than DP ($p < 0.02$), and has comparable results to Marginal Matching.

Systems	Science	Medical
baseline	22.20	25.32
DP-tripartite	22.76	25.81
PPDB$_{\text{fr}}$ (L)-tripartite	22.97	**27.11**
Marginal Matching	**23.62**	26.97

Table 6: BLEU scores for domain adaptation.

Systems	BLEU
baseline	29.59
DP-tripartite	30.08
PPDB$_{\text{arabic}}$ (L)-tripartite	**31.12**

Table 7: BLEU score results for Arabic-English.

Marginal Matching performs better in science domain but graph-propagation approach with PPDB outperforms it in medical domain getting a $+1.79$ BLEU score improvement over the baseline.

5.5 Case 3: Morphologically Rich Languages

Both Distribution Profiling and Bilingual Pivoting propose morphological variants of a word as paraphrase pairs. Even more so in PPDB due to pivoting over English. We choose Arabic-English task for this experiment. We train the SMT system on 685K sentence pairs (randomly selected from LDC2007T08 and LDC2008T09) and use NIST OpenMT 2012 for dev and test data. Arabic side of 1M sentences of LDC2007T08 and LDC2008T09 is used to extract unigram paraphrases for DP. Table 7 shows that PPDB (large; with phrases) resulted in $+1.53$ BLEU score improvement over DP which only slightly improved over baseline.

6 Related Work

Sentence level paraphrasing has been used for generating alternative reference translations (Madnani et al., 2007; Kauchak and Barzilay, 2006), or augmenting the training data with sentential paraphrases (Bond et al., 2008; Nakov, 2008; Mirkin et al., 2009). Phrase level paraphrasing was done using crowdsourcing (Resnik et al., 2010) or by using paraphrases in lattice decoding (Onishi et al., 2010; Du et al., 2010).

Daumé and Jagarlamudi (2011) apply a generative model to domain adaptation based on canonical correlation analysis Haghighi et al. (2008). However, they use artificially created monolingual corpora very related to the same domain as test data. Irvine and Callison-Burch (2014a) generate a large, noisy phrase table by composing unigram translations which are obtained by a supervised method (Irvine and Callison-Burch, 2013). Comparable monolingual data is used to re-score and filter the phrase table. Zhang and Zong (2013) use a large manually generated lexicon for domain adaptation. In contrast to these methods, our method is unsupervised.

Alexandrescu and Kirchhoff (2009) use a graph-based semi-supervised model determine similarities between sentences, then use it to rerank the n-best translation hypothesis. Liu et al. (2012) extend this model to derive some features to be used during decoding. These approaches are orthogonal to our approach. Saluja et al. (2014) use Structured Label Propagation (Liu et al., 2012) in two parallel graphs constructed on source and target paraphrases. In their case the graph construction is extremely expensive. Leveraging a morphological analyzer, they reach significant improvement on Arabic. We can not directly compare our results to (Saluja et al., 2014) because they exploit several external resources such as a morphological analyzer and also had different sizes of training and test. In experiments (Sec. 5) we obtained comparable BLEU score improvement on Arabic-English by using bilingual pivoting only on source phrases. (Saluja et al., 2014) also use methods similar to (Habash, 2008) that expand the phrase table with spelling and morphological variants of OOVs in test data. We do not use the dev/test data to augment the phrase table.

Using comparable corpora to extract parallel sentences and phrases (Munteanu and Marcu, 2006; Smith et al., 2010; Tamura et al., 2012) are orthogonal to the approach we discuss here.

Bilingual and multilingual word and phrase representation using neural networks have been applied to machine translation (Zou et al., 2013; Mikolov et al., 2013a; Zhang et al., 2014). However, most of these methods focus on frequent words or an available bilingual phrase table (Zou et al., 2013; Zhang et al., 2014; Gao et al., 2014). Mikolov et al. (2013a) learn a global linear projection from source to target using representation of frequent words on both sides. This model can be used to generate translations for new words, but a large amounts of bilingual data is required to create such a model. (Mikolov et al., 2013b) also uses bilingual data to project new translation rules. Zhao et al. (2015) extend Mikolov's model to learn one local linear projection for each phrase. Their model reaches comparable results to Saluja et al. (2014) while works faster. Alkhouli et al. (2014) use neural network phrase representation for paraphrasing OOVs and find translation for them using a phrase-table created from limited parallel data. Our experimental settings is different from the approaches in (Alkhouli et al., 2014; Mikolov et al., 2013a; Mikolov et al., 2013b).

7 Conclusion and Future work

In future work, we would like to include translations for infrequent phrases which are not OOVs. We would like to explore new propagation methods that can directly use confidence estimates and control propagation based on label sparsity. We also would like to expand this work for morphologically rich languages by exploiting other resources like morphological analyzer and campare our approach to the current state of art approaches which are using these types of resources. In conclusion, we have shown significant improvements to the quality of statistical machine translation in three different cases: low resource SMT, domain shift, and morphologically complex languages. Through the use of semi-supervised graph propagation, a large scale multilingual paraphrase database can be used to improve the quality of statistical machine translation.

Acknowledgments

The authors would like to thank Chris Callison-Burch and Juri Ganitkevitch for providing us the latest version of PPDB, the anonymous reviewers for their comments. The research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC RGPIN 262313 and RGPAS 446348) to the last author.

References

Andrei Alexandrescu and Katrin Kirchhoff. 2009. Graph-based learning for statistical machine translation. In *NAACL 2009*.

Tamer Alkhouli, Andreas Guta, and Hermann Ney. 2014. Vector space models for phrase-based machine translation. In *EMNLP 2014: Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *ACL 2005*.

Francis Bond, Eric Nichols, Darren Scott Appling, and Michael Paul. 2008. Improving statistical machine translation by paraphrasing the training data. In *IWSLT 2008*.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *NAACL 2006*.

Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *EMNLP 2008*.

Marine Carpuat, H Daumé III, Alexander Fraser, Chris Quirk, Fabienne Braune, Ann Clifton, Ann Irvine, Jagadeesh Jagarlamudi, John Morgan, Majid Razmara, Aleš Tamchyna, Katharine Henry, and Rachel Rudinger. 2012. Domain adaptation in machine translation: Final report. In *2012 Johns Hopkins Summer Workshop*.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In *ACL 2011*.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research*.

Hal Daumé, III and Jagadeesh Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *ACL 2011*.

Jinhua Du, Jie Jiang, and Andy Way. 2010. Facilitating translation using source language paraphrase lattices. In *EMNLP 2010*.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *ACL 2010*.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *NAACL HLT 2013*.

Juri Ganitkevitch and Chris Callison-Burch. 2014. The multilingual paraphrase database. In *LREC 2014*.

Juri Ganitkevitch, Chris Callison-Burch, Courtney Napoles, and Benjamin Van Durme. 2011. Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation. In *NAACL HLT 2011*.

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2014. Learning continuous phrase representations for translation modeling. In *ACL 2014*.

Nikesh Garera, Chris Callison-Burch, and David Yarowsky. 2009. Improving translation lexicon induction from monolingual corpora via dependency contexts and part-of-speech equivalences. In *CoNLL 2009*.

Nizar Habash. 2008. Four Techniques for Online Handling of Out-of-Vocabulary Words in Arabic-English Statistical Machine Translation. In *ACL 2008*.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *ACL 2008*.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *WMT 2011*.

Ann Irvine and Chris Callison-Burch. 2013. Supervised bilingual lexicon induction with multiple monolingual signals. In *NAACL 2013*.

Ann Irvine and Chris Callison-Burch. 2014a. Hallucinating phrase translations for low resource MT. *CoNLL-2014*.

Ann Irvine and Chris Callison-Burch. 2014b. Using comparable corpora to adapt mt models to new domains. *ACL 2014*.

Ann Irvine, Chris Quirk, and Hal Daumé III. 2013. Monolingual marginal matching for translation model adaptation. In *EMNLP 2013*.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *NAACL 2008*.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *ACL 2002 workshop on unsupervised lexical acquisition*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL 2003*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, and et al. 2007. Moses: open source toolkit for statistical machine translation. In *ACL 2007*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit 2005*, volume 5.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *ACL 1998*.

Shujie Liu, Chi-Ho Li, Mu Li, and Ming Zhou. 2012. Learning translation consensus with structured label propagation. In *ACL 2012*.

Nitin Madnani, Necip Fazil Ayan, Philip Resnik, and Bonnie J Dorr. 2007. Using paraphrases for parameter tuning in statistical machine translation. In *WMT 2007*.

Gideon S. Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *NAACL 2001*.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *EMNLP 2009*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *CoRR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS 2013*.

Shachar Mirkin, Lucia Specia, Nicola Cancedda, Ido Dagan, Marc Dymetman, and Idan Szpektor. 2009. Source-language entailment modeling for translating unknown terms. In *ACL-IJCNLP 2009*.

Behrang Mohit, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A Smith. 2012. Recall-oriented learning of named entities in arabic wikipedia. In *EACL 2012*, pages 162–173.

Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting parallel sub-sentential fragments from non-parallel corpora. In *ACL 2006*.

Preslav Nakov. 2008. Improved statistical machine translation using monolingual paraphrases. In *ECAI 2008: 18th European Conference on Artificial Intelligence*. IOS Press.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *ACL 2003*.

Takashi Onishi, Masao Utiyama, and Eiichiro Sumita. 2010. Paraphrase lattice for statistical machine translation. In *ACL 2010*.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *ACL 1995*.

Majid Razmara, Maryam Siahbani, Reza Haffari, and Anoop Sarkar. 2013. Graph propagation for paraphrasing out-of-vocabulary words in statistical machine translation. In *ACL 2013*.

Philip Resnik, Olivia Buzek, Chang Hu, Yakov Kronrod, Alex Quinn, and Benjamin B Bederson. 2010. Improving translation via targeted paraphrasing. In *EMNLP 2010*.

Avneesh Saluja, Hany Hassan, Kristina Toutanova, and Chris Quirk. 2014. Graph-based semi-supervised learning of translation models from monolingual data. In *ACL 2014*.

Charles Schafer and David Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. In *CoNLL 2002*.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *NAACL 2010*.

Partha Pratim Talukdar and Koby Crammer. 2009. New Regularized Algorithms for Transductive Learning. In *European Conference on Machine Learning*.

Partha Pratim Talukdar, Joseph Reisinger, Marius Paşca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *EMNLP 2008*.

Partha Pratim Talukdar. 2009. Topics in graph construction for semi-supervised learning. Technical Report MS-CIS-09-13, University of Pennsylvania, Dept of Computer and Info. Sci.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2012. Bilingual lexicon extraction from comparable corpora using label propagation. In *EMNLP-CoNLL 2012*.

Jörg Tiedemann. 2009. News from OPUS-A collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*.

Jiajun Zhang and Chengqing Zong. 2013. Learning a phrase-based translation model from monolingual data with application to domain adaptation. In *ACL 2013*.

Jiajun Zhang, Feifei Zhai, and Chengqing Zong. 2012. Handling unknown words in statistical machine translation from a new perspective. In *Natural Language Processing and Chinese Computing*. Springer.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *ACL 2014*.

Shiqi Zhao, Haifeng Wang, Ting Liu, and Sheng Li. 2008. Pivot approach for extracting paraphrase patterns from bilingual corpora. In *ACL 2008*.

Kai Zhao, Hany Hassan, and Michael Auli. 2015. Learning translation models from monolingual continuous representations. In *NAACL 2015*.

Will Zou, Richard Socher, Daniel Cer, and Christopher Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP 2013*.