Graph-Based Collective Lexical Selection for Statistical Machine Translation

Jinsong Su[1,2], **Deyi Xiong**[3]*, **Shujian Huang**[2], **Xianpei Han**[4], **Junfeng Yao**[1]
Xiamen University, Xiamen, P.R. China[1]
State Key Laboratory for Novel Software Technology, Nanjing University, P.R. China[2]
Soochow University, Suzhou, P.R. China[3]
Institute of Software, Chinese Academy of Sciences, Beijing, P.R. China[4]
{jssu, yao0010}@xmu.edu.cn huangsj@nlp.nju.edu.cn
dyxiong@suda.edu.cn xianpei@nfs.iscas.ac.cn

Abstract

Lexical selection is of great importance to statistical machine translation. In this paper, we propose a graph-based framework for collective lexical selection. The framework is established on a *translation graph* that captures not only local associations between source-side content words and their target translations but also target-side global dependencies in terms of relatedness among target items. We also introduce a random walk style algorithm to collectively identify translations of source-side content words that are strongly related in translation graph. We validate the effectiveness of our lexical selection framework on Chinese-English translation. Experiment results with large-scale training data show that our approach significantly improves lexical selection.

1 Introduction

Lexical selection, which selects appropriate translations for lexical items on the source side, is a crucial task in statistical machine translation (SMT). The task is closely related to two factors: 1) associations of selected translations with lexical items on the source side, including corresponding source items and their neighboring words, and 2) dependencies[1] between selected target translations and other items on the target side.

As translation rules and widely-used n-gram language models can only capture local associations and dependencies, we have witnessed increasing efforts that attempt to incorporate non-local associations/dependencies into lexical selection. Efforts using source-side associations mainly focus on the exploitation of either sentence-level context (Chan et al., 2007; Carpuat and Wu, 2007; Hasan et al., 2008; Mauser et al., 2009; He et al., 2008; Shen et al., 2009) or the utilization of document-level context (Xiao et al., 2011; Ture et al., 2012; Xiao et al., 2012; Xiong et al., 2013). In contrast, target-side dependencies attract little attention, although they have an important impact on the accuracy of lexical selection. The most common practice is to use language models to estimate the strength of target-side dependencies (Koehn et al., 2003; Shen et al., 2008; Xiong et al., 2011). However, conventional n-gram language models are not good at capturing long-distance dependencies. Consider the example shown in Figure 1. As the translations of polysemous words "*wèntí*", "*chíyǒu*" and "*lìchǎng*" are far from each other, our baseline can only correctly translate "*lìchǎng*" as "*stance*". It inappropriately translates the other two words as "*problem*" and *null*, respectively, even with the support of an n-gram language model. If we could model long-distance dependencies among target translations of source words "*wèntí*"(*issue*), "*chíyǒu*"(*hold*) and "*lìchǎng*"(*stance*), these translation errors could be avoided.

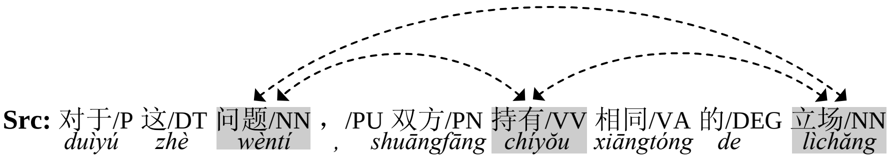
In order to model target-side global dependencies, we propose a novel graph-based collective lexical selection framework for SMT. Specifically,

- First, we propose a translation graph to model not only local associations between source-side content words and their target translations but also global relatedness among target-side items.

*Corresponding author.

[1] Please note that dependencies in this paper are not necessarily syntactic dependencies.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1238–1247,
Lisbon, Portugal, 17-21 September 2015.


Src: 对于/P 这/DT 问题/NN ，/PU 双方/PN 持有/VV 相同/VA 的/DEG 立场/NN
duìyú zhè wèntí , shuāngfāng chíyǒu xiāngtóng de lìchăng

wèntí = {*problem*, *question*, *issue* ...}
chíyǒu = {*hold*, *null*, *possess* ...}
lìchăng = {*stance*, *position* ...}

Tran: *For this* ***problem*** *, two sides of the same or similar* ***stance*** *.*
Ref: *Both sides* ***hold*** *the same or similar* ***position*** *on this* ***issue*** *.*

Figure 1: A Chinese-English translation example to illustrate the importance of target-side long-distance dependencies for lexical selection. Dotted lines show long-distance dependencies of source content words. Three content words "*wèntí*", "*chíyǒu*", "*lìchăng*", and their candidate translations with high translation probabilities are also presented. **Src**: A Chinese sentence with part-of-speech tags. **Tran**: system output. **Ref**: reference translation.

- Second, we introduce a collective lexical selection algorithm, which can jointly identify translations of all source-side content words in the translation graph.

- Finally, we incorporate confidence scores of candidate translations in the translation graph, which are derived by the collective selection algorithm, into SMT to improve lexical selection.

We validate the effectiveness of our graph-based lexical selection framework on a hierarchical phrase-based system (Chiang, 2007). Experiment results on the NIST Chinese-English test sets show that our approach significantly improves translation quality.

We begin in Section 2 with the construction of translation graph for each translated sentence. Then, we propose a graph-based collective lexical selection framework for SMT in Section 3. Experiment results are reported in Section 4. We summarize and compare related work in Section 5. Finally, Section 6 presents conclusions and directions for future research.

2 Translation Graph

Formally, a translation graph is a weighted graph G=(N, E). In the node set N, each node represents either a source word or a target translation that contains one or multiple target words. In the edge set E, an edge linking a source word to a target translation is referred to as a *source-target association edge*, and an edge connecting two target translations is called as a *target-target relatedness edge*. In Section 2.1 and 2.2, we will answer the following two questions on the translation graph:

- Which source words and their translations should be included in the translation graph?

- How can we measure the strength of the above two types of relations in the graph with edge weights?

2.1 Graph Nodes

For a source sentence, the most ideal translation graph is a graph that includes all source words and their candidate translations. However, this ideal graph has two problems: intensive computation for graph inference and difficulty in modeling dependencies between function and content words. In order to get around these two issues, we only consider lexical selection for source content words[2].

We first identify source-side content word pairs using statistical metrics, and then keep word pairs with a high relatedness strength in the translation graph. To be specific, we use pointwise mutual information (PMI) (Church and Hanks, 1990) and co-occurrence frequency to measure the relatedness strength of two source-side words s and s' within a window d_s. Content word pairs will be kept when their co-occurrence frequencies are more than ϵ_{cf} times in our training corpus and PMI values are larger than ϵ_{pmi}. In this process, we remove noisy word pairs using the following heuristic rules: (1) As an adjective only has relations with its head nouns or dependent adverbs, we remove all word pairs where an adjective is paired with words other than its head nouns or dependent adverbs; (2) We apply a similar constraint to adverbs too, since the same thing happens to an adverb and its head verb or head ad-

[2] In this work, we consider nouns, verbs, adjectives and adverbs as content words in the source/target language.

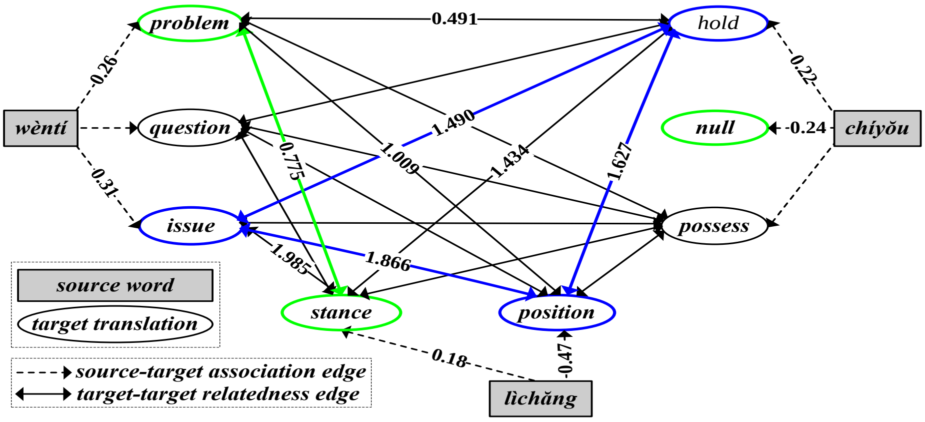


Figure 2: Translation graph of the example shown in Figure 1. Relatedness scores on edges are shown for two group of translations {"*problem*", *null*, "*stance*"} (**green**) and {"*issue*", "*hold*", "*position*"} (**blue**), which are estimated with PMI. Note that the *null* node does not have any relations with other nodes. Besides, two translation combinations {"*problem*", *null*, "*stance*"} (green) and {"*issue*", "*hold*", "*position*"} (blue) have different strengths of relatedness.

jective. For example, in the Chinese sentence in Figure 1, the adjective "*xiāngtóng*" is only related to the noun "*lìchǎng*" although it also frequently co-occur with "*wèntí*".

After identifying source-side content word pairs, we collect all target translations of these content words from extracted bilingual rules according to word alignments. These content words and target translations are used to build a translation graph, where each node represents a source-side content word or a candidate target translation. Note that there may be hundreds of different translations for a source word. For simplicity, we only consider target translations from translation options that are adopted by the decoder after rule filtering. Let's revisit the example in Figure 2, we include the following target translations in the translation graph: "*problem*", "*question*", "*issue*", "*hold*", *null*, "*possess*", "*stance*" and "*position*".

2.2 Edges and Weights

In this section, we introduce how we calculate weights for two kinds of edges in a translation graph.

2.2.1 Source-Target Association Edge

Connecting a source-side content word and its candidate target translations, a source-target association edge provides a way to propagate translation association evidence from a source word to its candidate translations. Obviously, the stronger the association between a source word and its candidate translation, the more evidence the corresponding edge will propagate. For each source-side content word, we obtain its candidate translations via the kept word alignments. Following Xiong et al. (2014), we allow a target translation to be either a phrase of length up to 3 words or *null* when s is not aligned to any word in the corresponding bilingual rule. We define the weight of the edge from a source-side content word s to its target translation $\tilde{t}$ as follows:

$$Weight(s \rightarrow \tilde{t}) = \frac{TP(s,\tilde{t})}{\sum_{\tilde{t}' \in N(s)} TP(s,\tilde{t}')} \qquad (1)$$

where $N(s)$ denotes the set of candidate target translations of s kept on the translation graph, and $TP(s,\tilde{t})$ measures the probability of s being translated to $\tilde{t}$. It is very important to note that there is no evidence propagated from a target translation to a source word, as source-target association edges only go from a source-side content word to its translations.

We compute $TP(s,\tilde{t})$ according to the principle of maximal likelihood as follows:

$$TP(s,\tilde{t}) = \frac{count(s,\tilde{t})}{count(s)} \qquad (2)$$

where $count(s,\tilde{t})$ indicates how often s is aligned to $\tilde{t}$ in the training corpus. Using this method, we can compute the translation probabilities of the *source-target association edges* in Figure 2 as follows: *TP*("*wèntí*", "*issue*")=0.31, *TP*("*chíyǒu*", "*hold*")=0.22 and *TP*("*lìchǎng*", "*position*")=0.47.

2.2.2 Target-Target Relatedness Edge

Connecting two target translations of different related source content words, a target-target relatedness edge enables translation graph to capture dependencies between translations of any two different source words.

Computing the weight of a target-target relatedness edge is crucial for our method. Intuitively, the stronger co-occurrence strength two translations have, the more evidence should be propagated between them. Therefore we calculate the weight of a target-target related edge based on the co-occurrence strength of two translations linked by the edge. Formally, given the translation $\tilde{t}$ of source-side content word s and the translation $\tilde{t}'$ of source-side content word s', the weight of the edge from $\tilde{t}$ to $\tilde{t}'$ is defined as follows:

$$Weight(\tilde{t} \rightarrow \tilde{t}') = \frac{RS(\tilde{t}, \tilde{t}')}{\sum_{\tilde{t}'' \in N(\tilde{t})} RS(\tilde{t}, \tilde{t}'')} \qquad (3)$$

where $N(\tilde{t})$ denotes the set of candidate translations that link to $\tilde{t}$, and $RS(\tilde{t}, \tilde{t}')$ measures the strength of relatedness between $\tilde{t}$ and $\tilde{t}'$ which is calculated as the average word-level relatedness over all content words in these two translations $\tilde{t}$ and $\tilde{t}'$.

As for the word-level relatedness $RS(t, t')$ for a content word pair (t, t'), we estimate it with the following two approaches over collected co-occurring word pairs within a window of size d_t: (1) $RS(t, t')$ is computed as a bigram conditional probability $p_{lm}(t'|t)$ via the language model; (2) Following (Xiong et al., 2011) and (Liu et al., 2014), we employ PMI to define $RS(t, t')$ as $\ln \frac{p(t,t')}{p(t)p(t')}$.

3 Collective Lexical Selection Algorithm

Based on the translation graph, we propose a collective lexical selection algorithm to jointly identify translations of all source words in the graph.

3.1 Problem Statement and Solution Method

As stated previously, the translation of a source-side content word s should be: 1) associated with s; 2) related to the translations of other source-side content words. Thus, in the translation graph, the translation of s should be a target-side node which has: 1) an association edge with the node of s; 2) many relatedness edges with other target-side nodes that represent translations of other source words.

Let's revisit Figure 2. If we know that the translation of "*wèntí*" is "*issue*", the relatedness between (*"issue", "hold"*) and between (*"issue", "position"*) can provide evidences that "*hold*" and "*position*" are the correct translations of "*chíyǒu*" and "*lìchǎng*", respectively. On the other hand, the candidate translation "*problem*" is less related to "*hold*" and "*position*", which may suggest that it is not likely to be the correct translation of "*wèntí*", even if it has a strong source-target association relation with "*wèntí*". However, in the translation graph, the correct target translation of a source word depends on correct translations of other source words in the graph, and vice versa. So how do we find these correct translations?

We propose a *Random Walk* (Gobel and Jagers, 1974) style algorithm to solve this problem, aiming to use both local source-target associations and global target-target relatedness simultaneously during translation. In our algorithm, we assign each node an *evidence score* in the translation graph, which indicates either the importance of a source word (for a source word node) or the confidence of a target translation being a correct translation (for a target word node). Specifically, we perform collective inference on the translation graph as follows:

- First, we set initial evidence scores for nodes in the translation graph.
- Second, evidence scores are simultaneously updated by propagating evidences along edges in the translation graph.

In the following sub-section, we describe the two steps in detail.

3.2 Details of Our Algorithm

Using the algorithm shown in Algorithm 1, we iteratively derive evidence scores for candidate translations.

3.2.1 Notations

For a translation graph with $\boldsymbol{n}$ nodes, we assign each node an index from 1 to $\boldsymbol{n}$ and use this index to represent the node. We also use the following two notations:

- **The evidence vector $\boldsymbol{V}$**: an n-dimensional vector where the ith component $\boldsymbol{V}_i$ is the evidence score contained in this node (if node i corresponds to a source word), or the evidence score from the related translations (if node i corresponds to a target translation). In particular, we use $\boldsymbol{V}^{(0)}$ to denote the initial

Algorithm 1 Collective Inference in Translation Graph.

```
Input: $S$: the set of source-side content words, and $S(i)$
         denotes the source word of node $i$;
       $k$: the number of source-side content words ;
       $T$: the set of all candidate target translations, and $T(j)$
         denotes the target translation of node $j$;
       $l$: the number of candidate target translations;
       $\lambda$: the reallocation weight;
       $maxIter$: the maximum iteration number;
       $\epsilon$: the difference threshold;
 1: for $i = 1, 2..., k$
 2:   for $j = 1, 2..., l$
 3:     if $S(i)$ is linked to $T(j)$
 4:       $M_{k+j,i} \leftarrow Weight(S(i) \rightarrow T(j))$
 5: for $j_1 = 1, 2..., l$
 6:   for $j_2 = 1, 2..., l$
 7:     if $T(j_1)$ is linked to $T(j_2)$
 8:       $M_{k+j_2,k+j_1} \rightarrow Weight(T(j_1) \rightarrow T(j_2))$
 9: for $i = 1, 2..., k$
10:   $V_i^{(0)} \leftarrow Importance(S(i))$
11: for $j = 1, 2..., l$
12:   $V_{k+j}^{(0)} \leftarrow 0$
13: $\delta \leftarrow \infty$
14: $r \leftarrow 1$
15: while $r \leq maxIter$ && $\delta > \epsilon$ do
16:   $V^{(r)} \leftarrow (1 - \lambda) \times M \times V^{(r-1)} + \lambda \times V^{(0)}$
17:   $\delta \leftarrow \|V^{(r)} - V^{(r-1)}\|_2$
18:   $r \leftarrow r + 1$
19: end while
20: for $i = 1, 2..., k$
21:   for $j = 1, 2..., l$
22:     if $S(i)$ is linked to $T(j)$
23:       $LexiTable(S(i), T(j)) \leftarrow normalize(V_{k+j}^{(r)})$
Return: $LexiTable$;
```

evidence vector, and $\boldsymbol{V}^{(r)}$ to represent the evidence vector we obtain at the rth iteration.

- **The evidence propagation matrix** $\boldsymbol{M}$: an $\boldsymbol{n} \times \boldsymbol{n}$ matrix where $\boldsymbol{M}_{ij}$ is the evidence propagation ratio from node j to node i, and its value is the weight of the edge from node j to node i.

3.2.2 Algorithm

In Algorithm 1, we jointly infer the evidence scores of all candidate translations in the following three steps.

In **Step 1**, we calculate the evidence propagation matrix $\boldsymbol{M}$ according to the method described in Section 2.2 (equations (1) and (3)) (**Lines 1-8**).

In **Step 2**, we adopt different methods to set the value of $\boldsymbol{V}^{(0)}$ according to the node type. If the node corresponds to a source word, we set the initial value using its importance score in the translation graph, as implemented in (Han et al. 2011) (**Lines 9-10**). We calculate the importance score of the source word s using *tf.idf* as follows:

$$\textbf{\textit{Importance}}(s) = \frac{tf.idf(s)}{\sum_{s' \in N_{src}} tf.idf(s')} \tag{4}$$

where N_{src} is the set of source words in the translation graph. If the node corresponds to a target translation, its initial evidence score is 0 (**Lines 11-12**).

In **Step 3**, evidences are simultaneously reinforced by propagating them among semantically related translations (**Lines 13-19**). Specific to our algorithm, we update them by propagating evidences according to different types of relations in the evidence propagation matrix $\boldsymbol{M}$. Formally, the recursive update of the evidence vector is defined as follows:

$$\boldsymbol{V}^{(r)} = \boldsymbol{M} \times \boldsymbol{V}^{(r-1)} \tag{5}$$

where r is the number of iterations.

One problem with the above equation is that some nodes in the translation graph do not have evidence outgoing edges, such as translation nodes containing only function words or the *null* node. The evidence will disappear when passing through these nodes. To solve this problem, we propagate evidence in the form of reallocation: we reallocate a fraction of evidence to the initial evidence vector $\boldsymbol{V}^{(0)}$ at each step. The new recursive update of the evidence vector is formulated as follows:

$$\boldsymbol{V}^{(r)} = (1 - \lambda) \times \boldsymbol{M} \times \boldsymbol{V}^{(r-1)} + \lambda \times \boldsymbol{V}^{(0)} \tag{6}$$

where $\lambda \in (0, 1)$ is the fraction of the reallocated evidence. We keep updating the evidence vector according to this equation (**Line 16**), until the maximal number of iteration $maxIter$ is reached or the Euclidean distance (**Line 17**) between evidence vectors calculated in two consecutive iterations is less than a pre-defined threshold ϵ (**Line 15**).

In this way, we jointly infer the evidence scores of all candidate target translations in the translation graph. Table 1 gives the evidence scores of the example in Figure 2. We can find that our system enhanced with target translation dependencies is able to select correct translations.

3.2.3 Integration of Derived Evidence Score

For each translated sentence, we may build multiple translation graphs. For each translation graph,

	scores of the source words			scores of the target translations							
	wèntí	*chíyǒu*	*lìchǎng*	*problem*	*question*	*issue*	*hold*	*null*	*possess*	*stance*	*position*
$V^{(0)}$	0.2015	0.3989	0.3996	0	0	0	0	0	0	0	0
$V^{(r)}$	0.0302	0.0598	0.0599	0.0533	0.0774	**0.1071**	**0.1218**	0.0244	0.0604	0.1186	**0.1486**

Table 1: The initial and final evidence scores of some source words and their target translations in Figure 2. Here we set the reallocation weight λ as 0.15. Note that the translations "*issue*", "*hold*" and "*position*" are given high evidence scores.

we infer evidence scores of translations represented by graph nodes using the above-mentioned algorithm before decoding. Then, for each candidate translation of a source-side content word, we normalize its evidence score over the corresponding translation graph to form an additional lexical translation probability (**Lines 20-23**). For instance, the normalized evidence score of "*chíyǒu*" translated into "*hold*" is calculated as $0.1218/(0.1218+0.0244+0.0604) \approx 0.5895$. In this way, for each bilingual rule with word alignments, we will obtain a new lexical weight which can be used together with the original translation probabilities and lexical weight to improve lexical selection in SMT.

4 Experiments

4.1 Setup

Our bilingual training corpus is the combination of the FBIS corpus and Hansards part of LDC2004T07 corpus ($1M$ parallel sentences, $54.6K$ documents, with $25.2M$ Chinese words and $29M$ English words). We word-aligned them using *GIZA++* (Och and Ney, 2003) with the option "*grow-diag-final-and*". We chose the NIST evaluation set of 2005 (MT05) as the development set, and the sets of MT06/MT08 as test sets. We used *SRILM* Toolkit (Stolcke, 2002) to train one 5-gram language model on the Xinhua portion of Gigaword corpus.

To construct translation graphs, we first used the *ZPar* toolkit[3] and the Stanford toolkit[4] to preprocess (word segmentation, PoS tagging and so on) Chinese and English sentences, respectively. We used the Chinese part of our bilingual corpus and an additional Chinese LDC Xinhua news corpus ($10.2M$ sentences with $279.9M$ words) as training data to collect Chinese word pairs. We set window size d_s=15, thresholds ϵ_{pmi}=0, ϵ_{cf}=5 to identify Chinese related word pairs in the NIST translated sentences. Averagely, these three sets contain 13.5, 10.3 and 9.5 content words used to build translation graphs per sentence, respectively. Using the English part of our bilingual corpus and the Xinhua portion of Gigaword corpus as training data, we set window size d_t=20, and used the SRILM toolkit with Witten-Bell smoothing and PMI to calculate relatedness strengths for target-side translations. To avoid data sparseness, we build the graph using the surface forms of words while calculating the word relatedness at the lemma level. To achieve this, we converted each word into its corresponding lemma with the exception of adjectives and adverbs. In the procedure of collective lexical selection, the difference threshold ϵ was set as 10^{-10}, and the maximal iteration number *maxIter* 100.

We reimplemented the decoder of Hiero (Chiang, 2007), a famous hierarchical phrase-based (HPB) system. HPB system is a formally syntax-based system and delivers good performance in various translation evaluations. During decoding, we set the *ttable-limit* as 20, the *stack-size* as 100. The translation quality is evaluated by case-insensitive BLEU-4 metric (Papineni et al., 2002). To alleviate the impact of the instability of MERT (Och, 2003), we ran it three times for each experiment and reported the average BLEU scores as suggested in (Clark et al., 2011). Finally, we conducted *paired bootstrap sampling* (Koehn, 2004) to test the significance in BLEU score differences.

4.2 Our Method vs Other Methods

In the first group of experiments, we investigated the effectiveness of our model by comparing it against the baseline as well as two additional models: (1) *lexicalized rule selection model* (He et al., 2008) (**LRSM**), which employs local context to improve rule selection in the HPB system; (2) *topic similarity model* (Xiao et al., 2012)[5] (**TSM**), which explores document-level topic information for translation rule selection in the HPB system. Furthermore, we combined our model with the two models to see if we could obtain further improvements. For this, we integrated the new lexi-

[3] http://people.sutd.edu.sg/~yue_zhang/doc/index.html

[4] http://nlp.stanford.edu/software

[5] We used 30 topics following (Xiao et al., 2012).

System	MT06	MT08	Avg
Baseline	30.25	21.25	25.75
LRSM	31.12	21.98	26.55
TSM	30.79	21.90	26.35
GM(LM)	30.64	21.78	26.21
GM(PMI)	31.02	21.77	26.40
LRSM +GM(PMI)	31.66	22.23	26.95
TSM +GM(PMI)	31.34	22.26	26.80

Table 2: Experiment results on the test sets with λ=0.15. **Avg** = average BLEU scores, **GM(LM)** and **GM(PMI)** denote our model using the measure based on language model and PMI, respectively.

cal weight learned by our model as a new feature into the **LRSM/TSM** system.

Table 2 reports the results. All models outperform the baseline. Especially, our graph-based lexical selection model GM(PMI) achieves an average BLEU score of **26.40** on the two test sets, which is higher than that of the baseline by **0.65** BLEU points. This improvement is statistically significant at p<0.01. The BLEU score of our model is close to those of LRSM and TSM, which achieve an average BLEU score of 26.55 and 26.35 on the two test sets, respectively. As PMI is slightly better than LM in our model, we use PMI in experiments hereafter.

The combination of our model and LRSM is able to further improve translation quality in terms of BLEU. In this case, the average BLEU score of the improved system is 26.95, with 0.4 BLEU points higher than LRSM. When combining our model with TSM, we obtain an average BLEU score of 26.80, which is better than TSM by 0.45 BLEU points. The two improvements over LRSM and TSM are also statistically significant at p<0.05. These experiment results suggest that exploring long-distance dependencies among target translations is complementary to the previous lexical selection methods which focus on source-side context information.

In order to know how our approach improves the performance of the HPB system, we compared the best translations of the HPB system using different models. We find that our approach really improves translation quality by utilizing target-side long-distance dependencies which are, on the contrary, ignored in previous methods.

For example, the source sentence "... 穆沙拉夫 去年 9月 和 [巴] 北方 地区... 塔利班 残余 势力 ..." is translated as follows:

- **Ref**: ... *musharraf and some tribal leaders in the northern region of [pakistan] last september ... the remnant forces of the taliban ...*

- **Baseline**: ... *musharraf last september and [palestine] north of tribal leaders ... the remnants of the taliban ...*

- **LRSM**: ... *musharraf last september and some tribal chiefs of the northern region of [palestine] ... the remnants of the taliban ...*

- **LRSM+GM(PMI)**: ... *last september musharraf and some tribal chiefs of the northern region of [pakistan] ... the remnants of the taliban ...*

Here both the baseline and LRSM fail to obtain the right translation for the word "巴" because "*palestine*" has a higher probability than "*pakistan*" (0.0374 vs 0.0285). However, in our model, the long-distance dependencies between (*"musharraf", "pakistan"*) and (*"taliban", "pakistan"*) help the decoder correctly choose the translation "*pakistan*" for "巴".

In yet another example, the source sentence "美 希望 朝 核 问题 [协议] 得到 全面 执行" is translated as follows:

- **Ref**: *us hopes agreement on north korean nuclear issue be fully implemented*

- **Baseline**: *us hoped that the dprk nuclear issue is the full implementation*

- **TSM**: *us hope that the full implementation of the nuclear issue*

- **TSM+GM(PMI)**: *us hope that the dprk nuclear issue [agreement] to be fully implemented*

Even with TSM, the HPB system did not translate "协议" at all because translation rules "X_1 协议 得到 ||| X_1 is" and "X_1 协议 X_2 执行 ||| X_2 implementation of X_1" are used to translate the source sentence by the baseline and TSM systems respectively. However, in the combined model TSM+GM(PMI), the differences in relatedness scores between (*"nuclear", "agreement"*), (*"issue", "agreement"*) and (*"agreement", "implemented"*) encourage the enhanced system to select right translation for this word.

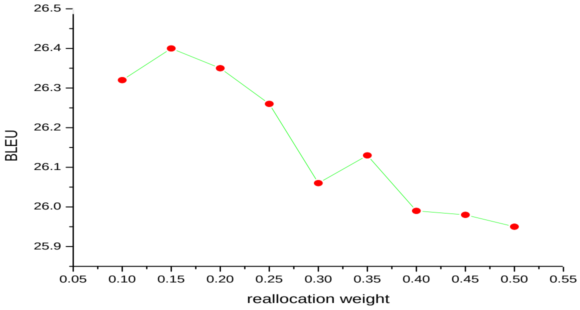


Figure 3: Experiment results on the test sets using different reallocation weights.

4.3 Effect of Reallocation Weight λ.

In Eq. (6), the reallocation weight λ determines which part plays a more important role in our method. In order to investigate the effect of λ on our method, we tried different values for λ: from 0.1 to 0.5 with an increment of 0.05 each time. The experimental setup is the same as the previous experiments. Figure 3 shows the average BLEU scores on the two test sets. Our system performs well when λ ranges from 0.1 to 0.25. The performance drops when λ is larger than 0.25. A small reallocation weight λ reduces the impact of initial evidences and local source-side associations in the collective lexical selection algorithm, but increases the impact of global dependencies of target-side translation, which are normally not considered in previous lexical selection methods. This performance curve on the values of λ suggests that target-side global dependencies are important for lexical selection.

5 Related Work

The collective inference algorithm is partially inspired by Han et al. (2011) who propose a graph-based collective entity linking (EL) method to model global interdependences among different EL decisions. We successfully adapt this algorithm to lexical selection in SMT. Other related work mainly includes the following two strands.

(1) Lexical selection in SMT. In order to capture source-side context for lexical selection, some researchers propose *trigger-based lexicon models* to capture long-distance dependencies (Hasan et al., 2008; Mauser et al., 2009), and many more researchers build classifiers with rich context information to select desirable translations during decoding (Chan et al., 2007; Carpuat and Wu, 2007; He et al., 2008; Liu et al., 2008). Shen et al. (2009) introduce four new linguistic and contextual features for HPB system. We have also witnessed increasing efforts in the exploitation of document-level context information. Xiao et al. (2011) impose a hard constraint to guarantee the translation consistency in document-level translation. Ture et al. (2012) soften this consistency constraint by integrating three counting features into decoder. Hardmeier et al. (2012, 2013) introduce a document-wide phrase-based decoder and integrate a semantic language model that cross sentence boundaries into the decoder. Based on topic models, Xiao et al. (2012) present a topic similarity model for HPB system, where each rule is assigned with a topic distribution. Also relevant is the work of Xiong et al. (2013), who use three different models to capture lexical cohesion for document-level machine translation. Compared with the above-mentioned studies, our method focuses on the exploitation of global dependencies among target translations, which has attracted little attention before.

Different from exploring source-side context, other researchers pay attention to the utilization of target-side context information. The common practice in SMT is to use an n-gram language model to capture local dependencies between translations (Koehn et al., 2003; Xiong et al., 2011). Yet another approach exploring target-side context information is proposed by Shen et al. (2008), who use a dependency language model to capture long-distance relations on the target side. Moreover, Zhang et al. (2014) treat translation as an unconstrained target sentence generation task, using soft features to capture lexical and syntactic correspondences between the source and target language. Recently, many researcher have proposed to use deep neural networks to model long-distance dependencies of arbitrary length for SMT (Auli et al., 2013; Kalchbrenner and Blunsom, 2013; Devlin et al., 2014; Hu et al., 2014; Liu et al., 2014; Sundermeyer et al., 2014). Our work is significantly different from these methods. We use a graph representation to capture local and global context information, which, to the best of our knowledge, is the first attempt to explore graph-based representations for lexical selection. Furthermore, our model do not resort to any syntactic resources such as dependency parsers of the target language.

(2) Random walk for SMT. Because of the advantage of global consistency, random walk algorithm has been applied in SMT. For example, Cui et al. (2013) develop an effective approach to optimize phrase scoring and corpus weighting jointly using graph-based random walk. Zhu et al. (2013) apply a random walk method to discover implicit relations between the phrases of different languages. Aiming to better evaluate translation quality at the document level, Gong and Li (2013) run PageRank algorithm to assign weights to words in translation evaluation. Different from these studies, the key interest of our research lies in the lexical selection with random walk.

6 Conclusion and Future Work

This paper has presented a novel graph-based collective lexical selection method for SMT. We build translation graphs to capture local source-side associations and global target-side dependencies, and propose a purely collective inference algorithm to jointly identify target translations of source-side content words in translation graphs. Our method capitalizes on capabilities of translation graphs to represent both local and global relations on the source/target side. Experiment results demonstrate the effectiveness of our method.

In the future, we plan to further improve our model by capturing semantic relatedness among source words. Additionally, we also want to jointly model different levels of context information in a unified framework for SMT.

Acknowledgments

The authors were supported by National Natural Science Foundation of China (Grant Nos 61303082 and 61403269), Natural Science Foundation of Jiangsu Province (Grant No. BK20140355), Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20120121120046), Research fund of the State Key Laboratory for Novel Software Technology in Nanjing University (Grant No. KFKT2015B11), the Special and Major Subject Project of the Industrial Science and Technology in Fujian Province 2013 (Grant No. 2013HZ0004-1), and 2014 Key Project of Anhui Science and Technology Bureau (Grant No. 1301021018). We also thank the anonymous reviewers for their insightful comments.

References

Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. 2013. Joint language and translation modeling with recurrent neural networks. In *Proc. of EMNLP 2013*, pages 1044–1054.

Marine Carpuat and Dekai Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proc. of EMNLP 2007*, pages 61–72.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proc. of ACL 2007*, pages 33–40.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, pages 201–228.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proc. of ACL 2011, short papers*, pages 176–181.

Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li, and Ming Zhou. 2013. Bilingual data cleaning for smt using graph-based random walk. In *Proc. of ACL 2013*, pages 340–345.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proc. of ACL 2014*, pages 1370–1380.

F. Gobel and A.A. Jagers. 1974. Random walks on graphs. *Stochastic Processes and Their Applications*, 2(4):331–336.

Zhengxian Gong and Liangyou Li. 2013. Document-level automatic machine translation evaluation based on weighted lexical cohesion. In *Proc. of NLPCC 2013*.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: A graph-based method. In *Proc. of SIGIR 2011*, pages 765–774.

Saša Hasan, Juri Ganitkevitch, Hermann Ney, and Jesús Andrés-Ferrer. 2008. Triplet lexicon models for statistical machine translation. In *Proc. of EMNLP 2008*, pages 372–381.

Zhongjun He, Qun Liu, and Shouxun Lin. 2008. Improving statistical machine translation using lexicalized rule selection. In *Proc. of COLING 2008*, pages 321–328.

Yuening Hu, Michael Auli, Qin Gao, and Jianfeng Gao. 2014. Minimum translation modeling with recurrent neural networks. In *Proc. of EACL 2014*, pages 20–29.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proc. of EMNLP 2013*, pages 1700–1709.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL-HLT 2003*, pages 127–133.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP 2004*, pages 388–395.

Qun Liu, Zhongjun He, Yang Liu, and Shouxun Lin. 2008. Maximum entropy based rule selection model for syntax-based statistical machine translation. In *Proc. of EMNLP 2008*, pages 89–97.

Kang Liu, Liheng Xu, and Jun Zhao. 2014. Extracting opinion targets and opinionwords from online reviews with graph co-ranking. In *Proc. of ACL 2014*, pages 314–324.

Arne Mauser, Sas̆a Hasan, and Hermann Ney. 2009. Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Proc. of EMNLP 2009*, pages 210–218.

Franz Joseph Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51.

Franz Joseph Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL 2003*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proc. of ACL 2002*, pages 311–318.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proc. of ACL 2008*, pages 577–585.

Libin Shen, Jinxi Xu, Bing Zhang, Spyros Matsoukas, and Ralph Weischedel. 2009. Effective use of linguistic and contextual information for statistical machine translation. In *Proc. of EMNLP 2009*, pages 72–80.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proc. of ICSLP 2002*, pages 901–904.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. 2014. Translation modeling with bidirectional recurrent neural networks. In *Proc. of EMNLP 2014*, pages 14–25.

Ferhan Ture, DouglasW. Oard, and Philip Resnik. 2012. Encouraging consistent translation choices. In *Proc. of NAACL-HLT 2012*, pages 417–426.

Tong Xiao, Jingbo Zhu, Shujie Yao, and Hao Zhang. 2011. Document-level consistency verification in machine translation. In *Proc. of MT SUMMIT 2011*, pages 131–138.

Xinyan Xiao, Deyi Xiong, Min Zhang, Qun Liu, and Shouxun Lin. 2012. A topic similarity model for hierarchical phrase-based translation. In *Proc. of ACL 2012*, pages 750–758.

Deyi Xiong and Min Zhang. 2014. A sense-based translation model for statistical machine translation. In *Proc. of ACL 2014*, pages 1459–1469.

Deyi Xiong, Min Zhang, and Haizhou Li. 2011. Enhancing language models in statistical machine translation with backward n-grams and mutual information triggers. In *Proc. of ACL 2011*, pages 1288–1297.

Deyi Xiong, Guosheng Ben, Min Zhang, Yajuan L*ü*, and Qun Liu. 2013. Modeling lexical cohesion for document-level machine translation. In *Proc. of IJCAI 2013*, pages 2183–2189.

Yue Zhang, Kai Song, Linfeng Song, Jingbo Zhu, and Qun Liu. 2014. Syntactic smt using a discriminative text generation model. In *Proc. of EMNLP 2014*, pages 177–182.

Xiaoning Zhu, Zhongjun He, Hua Wu, Haifeng Wang, Conghui Zhu, and Tiejun Zhao. 2013. Improving pivot-based statistical machine translation using random walk. In *Proc. of EMNLP 2013*, pages 524–534.