Bilingual Data Cleaning for SMT using Graph-based Random Walk*

Lei Cui[†], **Dongdong Zhang**[‡], **Shujie Liu**[‡], **Mu Li**[‡], and **Ming Zhou**[‡]

[†]School of Computer Science and Technology
Harbin Institute of Technology, Harbin, China
leicui@hit.edu.cn
[‡]Microsoft Research Asia, Beijing, China
{dozhang,shujliu,muli,mingzhou}@microsoft.com

Abstract

The quality of bilingual data is a key factor in Statistical Machine Translation (SMT). Low-quality bilingual data tends to produce incorrect translation knowledge and also degrades translation modeling performance. Previous work often used supervised learning methods to filter low-quality data, but a fair amount of human labeled examples are needed which are not easy to obtain. To reduce the reliance on labeled examples, we propose an unsupervised method to clean bilingual data. The method leverages the mutual reinforcement between the sentence pairs and the extracted phrase pairs, based on the observation that better sentence pairs often lead to better phrase extraction and vice versa. End-to-end experiments show that the proposed method substantially improves the performance in large-scale Chinese-to-English translation tasks.

1 Introduction

Statistical machine translation (SMT) depends on the amount of bilingual data and its quality. In real-world SMT systems, bilingual data is often mined from the web where low-quality data is inevitable. The low-quality bilingual data degrades the quality of word alignment and leads to the incorrect phrase pairs, which will hurt the translation performance of phrase-based SMT systems (Koehn et al., 2003; Och and Ney, 2004). Therefore, it is very important to exploit data quality information to improve the translation modeling.

Previous work on bilingual data cleaning often involves some supervised learning methods. Several bilingual data mining systems (Resnik and Smith, 2003; Shi et al., 2006; Munteanu and Marcu, 2005; Jiang et al., 2009) have a post-processing step for data cleaning. Maximum entropy or SVM based classifiers are built to filter some non-parallel data or partial-parallel data. Although these methods can filter some low-quality bilingual data, they need sufficient human labeled training instances to build the model, which may not be easy to acquire.

To this end, we propose an unsupervised approach to clean the bilingual data. It is intuitive that high-quality parallel data tends to produce better phrase pairs than low-quality data. Meanwhile, it is also observed that the phrase pairs that appear frequently in the bilingual corpus are more reliable than less frequent ones because they are more reusable, hence most good sentence pairs are prone to contain more frequent phrase pairs (Foster et al., 2006; Wuebker et al., 2010). This kind of mutual reinforcement fits well into the framework of graph-based random walk. When a phrase pair p is extracted from a sentence pair s, s is considered casting a vote for p. The higher the number of votes a phrase pair has, the more reliable of the phrase pair. Similarly, the quality of the sentence pair s is determined by the number of votes casted by the extracted phrase pairs from s.

In this paper, a PageRank-style random walk algorithm (Brin and Page, 1998; Mihalcea and Tarau, 2004; Wan et al., 2007) is conducted to iteratively compute the importance score of each sentence pair that indicates its quality: the higher the better. Unlike other data filtering methods, our proposed method utilizes the importance scores of sentence pairs as fractional counts to calculate the phrase translation probabilities based on Maximum Likelihood Estimation (MLE), thereby none of the bilingual data is filtered out. Experimental results show that our proposed approach substantially improves the performance in large-scale Chinese-to-English translation tasks.

*This work has been done while the first author was visiting Microsoft Research Asia.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 340–345,
Sofia, Bulgaria, August 4-9 2013.

2 The Proposed Approach

2.1 Graph-based random walk

Graph-based random walk is a general algorithm to approximate the importance of a vertex within the graph in a global view. In our method, the vertices denote the sentence pairs and phrase pairs. The importance of each vertex is propagated to other vertices along the edges. Depending on different scenarios, the graph can take directed or undirected, weighted or un-weighted forms. Starting from the initial scores assigned in the graph, the algorithm is applied to recursively compute the importance scores of vertices until it converges, or the difference between two consecutive iterations falls below a pre-defined threshold.

2.2 Graph construction

Given the sentence pairs that are word-aligned automatically, an *undirected*, *weighted* bipartite graph is constructed which maps the sentence pairs and the extracted phrase pairs to the vertices. An edge between a sentence pair vertex and a phrase pair vertex is added if the phrase pair can be extracted from the sentence pair. Mutual reinforcement scores are defined on edges, through which the importance scores are propagated between vertices. Figure 1 illustrates the graph structure. Formally, the bipartite graph is defined as:

$$G = (V, E)$$

where $V = S \cup P$ is the vertex set, $S = \{s_i | 1 \leq i \leq n\}$ is the set of all sentence pairs. $P = \{p_j | 1 \leq j \leq m\}$ is the set of all phrase pairs which are extracted from S based on the word alignment. E is the edge set in which the edges are between S and P, thereby $E = \{\langle s_i, p_j \rangle | s_i \in S, p_j \in P, \phi(s_i, p_j) = 1\}$.

$$\phi(s_i, p_j) = \begin{cases} 1 & \text{if } p_j \text{ can be extracted from } s_i \\ 0 & \text{otherwise} \end{cases}$$

2.3 Graph parameters

For sentence-phrase mutual reinforcement, a non-negative score $r(s_i, p_j)$ is defined using the standard TF-IDF formula:

$$r(s_i, p_j) = \begin{cases} \frac{PF(s_i,p_j) \times IPF(p_j)}{\sum_{p' \in \{p | \phi(s_i,p)=1\}} PF(s_i,p') \times IPF(p')} & \text{if } \phi(s_i, p_j) = 1 \\ 0 & \text{otherwise} \end{cases}$$

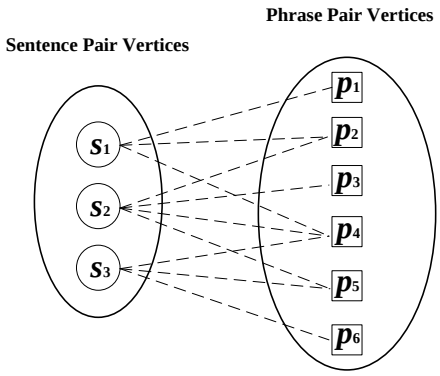

Figure 1: The circular nodes stand for S and square nodes stand for P. The lines capture the sentence-phrase mutual reinforcement.

where $PF(s_i, p_j)$ is the phrase pair frequency in a sentence pair and $IPF(p_j)$ is the inverse phrase pair frequency of p_j in the whole bilingual corpus. $r(s_i, p_j)$ is abbreviated as r_{ij}.

Inspired by (Brin and Page, 1998; Mihalcea and Tarau, 2004; Wan et al., 2007), we compute the importance scores of sentence pairs and phrase pairs using a PageRank-style algorithm. The weights r_{ij} are leveraged to reflect the relationships between two types of vertices. Let $u(s_i)$ and $v(p_j)$ denote the scores of a sentence pair vertex and a phrase pair vertex. They are computed iteratively by:

$$u(s_i) = (1-d) + d \times \sum_{j \in N(s_i)} \frac{r_{ij}}{\sum_{k \in M(p_j)} r_{kj}} v(p_j)$$

$$v(p_j) = (1-d) + d \times \sum_{j \in M(p_j)} \frac{r_{ij}}{\sum_{k \in N(s_i)} r_{ik}} u(s_i)$$

where d is empirically set to the default value 0.85 that is same as the original PageRank, $N(s_i) = \{j | \langle s_i, p_j \rangle \in E\}$, $M(p_j) = \{i | \langle s_i, p_j \rangle \in E\}$. The detailed process is illustrated in Algorithm 1. Algorithm 1 iteratively updates the scores of sentence pairs and phrase pairs (lines 10-26). The computation ends when difference between two consecutive iterations is lower than a pre-defined threshold δ (10^{-12} in this study).

2.4 Parallelization

When the random walk runs on some large bilingual corpora, even filtering phrase pairs that appear only once would still require several days of CPU time for a number of iterations. To overcome this problem, we use a distributed algorithm

Algorithm 1 Modified Random Walk

for all $i \in \{0 \dots |S|-1\}$ **do**
  $u(s_i)^{(0)} \leftarrow 1$
end for
for all $j \in \{0 \dots |P|-1\}$ **do**
  $v(p_j)^{(0)} \leftarrow 1$
end for
$\delta \leftarrow$ Infinity
$\epsilon \leftarrow$ threshold
$n \leftarrow 1$
while $\delta > \epsilon$ **do**
  for all $i \in \{0 \dots |S|-1\}$ **do**
    $F(s_i) \leftarrow 0$
    for all $j \in N(s_i)$ **do**
      $F(s_i) \leftarrow F(s_i) + \frac{r_{ij}}{\sum_{k \in M(p_j)} r_{kj}} \cdot v(p_j)^{(n-1)}$
    end for
    $u(s_i)^{(n)} \leftarrow (1-d) + d \cdot F(s_i)$
  end for
  for all $j \in \{0 \dots |P|-1\}$ **do**
    $G(p_j) \leftarrow 0$
    for all $i \in M(p_j)$ **do**
      $G(p_j) \leftarrow G(p_j) + \frac{r_{ij}}{\sum_{k \in N(s_i)} r_{ik}} \cdot u(s_i)^{(n-1)}$
    end for
    $v(p_j)^{(n)} \leftarrow (1-d) + d \cdot G(p_j)$
  end for
  $\delta \leftarrow \max(\triangle u(s_i)|_{i=1}^{|S|-1}, \triangle v(p_j)|_{j=1}^{|P|-1})$
  $n \leftarrow n+1$
end while
return $u(s_i)^{(n)}|_{i=0}^{|S|-1}$

based on the iterative computation in the Section 2.3. Before the iterative computation starts, the sum of the outlink weights for each vertex is computed first. The edges are randomly partitioned into sets of roughly equal size. Each edge $\langle s_i, p_j \rangle$ can generate two key-value pairs in the format $\langle s_i, r_{ij} \rangle$ and $\langle p_j, r_{ij} \rangle$. The pairs with the same key are summed locally and accumulated across different machines. Then, in each iteration, the score of each vertex is updated according to the sum of the normalized inlink weights. The key-value pairs are generated in the format $\langle s_i, \frac{r_{ij}}{\sum_{k \in M(p_j)} r_{kj}} \cdot v(p_j) \rangle$ and $\langle p_j, \frac{r_{ij}}{\sum_{k \in N(s_i)} r_{ik}} \cdot u(s_i) \rangle$. These key-value pairs are also randomly partitioned and summed across different machines. Since long sentence pairs usually extract more phrase pairs, we need to normalize the importance scores based on the sentence length. The algorithm fits well into the *MapReduce* programming model (Dean and Ghemawat, 2008) and we use it as our implementation.

2.5 Integration into translation modeling

After sufficient number of iterations, the importance scores of sentence pairs (i.e., $u(s_i)$) are obtained. Instead of simple filtering, we use the scores of sentence pairs as the fractional counts to re-estimate the translation probabilities of phrase pairs. Given a phrase pair $p = \langle \bar{f}, \bar{e} \rangle$, $A(\bar{f})$ and $B(\bar{e})$ indicate the sets of sentences that $\bar{f}$ and $\bar{e}$ appear. Then the translation probability is defined as:

$$P_{\text{CW}}(\bar{f}|\bar{e}) = \frac{\sum_{i \in A(\bar{f}) \cap B(\bar{e})} u(s_i) \times c_i(\bar{f}, \bar{e})}{\sum_{j \in B(\bar{e})} u(s_j) \times c_j(\bar{e})}$$

where $c_i(\cdot)$ denotes the count of the phrase or phrase pair in s_i. $P_{\text{CW}}(\bar{f}|\bar{e})$ and $P_{\text{CW}}(\bar{e}|\bar{f})$ are named as Corpus Weighting (CW) based translation probability, which are integrated into the log-linear model in addition to the conventional phrase translation probabilities (Koehn et al., 2003).

3 Experiments

3.1 Setup

We evaluated our bilingual data cleaning approach on large-scale Chinese-to-English machine translation tasks. The bilingual data we used was mainly mined from the web (Jiang et al., 2009)[1], as well as the United Nations parallel corpus released by LDC and the parallel corpus released by China Workshop on Machine Translation (CWMT), which contain around 30 million sentence pairs in total after removing duplicated ones. The development data and testing data is shown in Table 1.

Data Set	#Sentences	Source
NIST 2003 (dev)	919	open test
NIST 2005 (test)	1,082	open test
NIST 2006 (test)	1,664	open test
NIST 2008 (test)	1,357	open test
CWMT 2008 (test)	1,006	open test
In-house dataset 1 (test)	1,002	web data
In-house dataset 2 (test)	5,000	web data
In-house dataset 3 (test)	2,999	web data

Table 1: Development and testing data used in the experiments.

A phrase-based decoder was implemented based on inversion transduction grammar (Wu, 1997). The performance of this decoder is similar to the state-of-the-art phrase-based decoder in Moses, but the implementation is more straightforward. We use the following feature functions in the log-linear model:

[1] Although supervised data cleaning has been done in the post-processing, the corpus still contains a fair amount of noisy data based on our random sampling.

	dev	NIST 2005	NIST 2006	NIST 2008	CWMT 2008	IH 1	IH 2	IH 3
baseline	41.24	37.34	35.20	29.38	31.14	24.29	22.61	24.19
(Wuebker et al., 2010)	41.20	37.48	35.30	29.33	31.10	24.33	22.52	24.18
-0.25M	41.28	37.62	35.31	29.70	31.40	24.52	22.69	24.64
-0.5M	41.45	37.71	35.52	29.76	31.77	24.64	22.68	24.69
-1M	41.28	37.41	35.28	29.65	31.73	24.23	23.06	24.20
+CW	**41.75**	**38.08**	**35.84**	**30.03**	**31.82**	**25.23**	**23.18**	**24.80**

Table 2: BLEU(%) of Chinese-to-English translation tasks on multiple testing datasets ($p < 0.05$), where "-numberM" denotes we simply filter $number$ million low scored sentence pairs from the bilingual data and use others to extract the phrase table. "CW" means the corpus weighting feature, which incorporates sentence scores from random walk as fractional counts to re-estimate the phrase translation probabilities.

- phrase translation probabilities and lexical weights in both directions (4 features);
- 5-gram language model with Kneser-Ney smoothing (1 feature);
- lexicalized reordering model (1 feature);
- phrase count and word count (2 features).

The translation model was trained over the word-aligned bilingual corpus conducted by GIZA++ (Och and Ney, 2003) in both directions, and the diag-grow-final heuristic was used to refine the symmetric word alignment. The language model was trained on the LDC English Gigaword Version 4.0 plus the English part of the bilingual corpus. The lexicalized reordering model (Xiong et al., 2006) was trained over the 40% randomly sampled sentence pairs from our parallel data. Case-insensitive BLEU4 (Papineni et al., 2002) was used as the evaluation metric. The parameters of the log-linear model are tuned by optimizing BLEU on the development data using MERT (Och, 2003). Statistical significance test was performed using the bootstrap re-sampling method proposed by Koehn (2004).

3.2 Baseline

The experimental results are shown in Table 2. In the baseline system, the phrase pairs that appear only once in the bilingual data are simply discarded because most of them are noisy. In addition, the fix-discount method in (Foster et al., 2006) for phrase table smoothing is also used. This implementation makes the baseline system perform much better and the model size is much smaller. In fact, the basic idea of our "one count" cutoff is very similar to the idea of "leaving-one-out" in (Wuebker et al., 2010). The results show that the "leaving-one-out" method performs almost the same as our baseline, thereby cannot bring other benefits to the system.

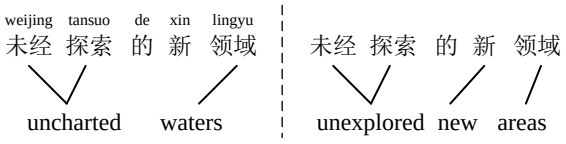

Figure 2: The left one is the non-literal translation in our bilingual corpus. The right one is the literal translation made by human for comparison.

3.3 Results

We evaluate the proposed bilingual data cleaning method by incorporating sentence scores into translation modeling. In addition, we also compare with several settings that filtering low-quality sentence pairs from the bilingual data based on the importance scores. The last N = { 0.25M, 0.5M, 1M } sentence pairs are filtered before the modeling process. Although the simple bilingual data filtering can improve the performance on some datasets, it is difficult to determine the border line and translation performance is fluctuated. One main reason is in the proposed random walk approach, the bilingual sentence pairs with non-literal translations may get lower scores because they appear less frequently compared with those literal translations. Crudely filtering out these data may degrade the translation performance. For example, we have a sentence pair in the bilingual corpus shown in the left part of Figure 2. Although the translation is correct in this situation, translating the Chinese word "*lingyu*" to "*waters*" appears very few times since the common translations are "*areas*" or "*fields*". However, simply filtering out this kind of sentence pairs may lead to some loss of native English expressions, thereby the trans-

lation performance is unstable since both non-parallel sentence pairs and non-literal but parallel sentence pairs are filtered. Therefore, we use the importance score of each sentence pair to estimate the phrase translation probabilities. It consistently brings substantial improvements compared to the baseline, which demonstrates graph-based random walk indeed improves the translation modeling performance for our SMT system.

3.4 Discussion

In (Goutte et al., 2012), they evaluated phrase-based SMT systems trained on parallel data with different proportions of synthetic noisy data. They suggested that when collecting larger, noisy parallel data for training phrase-based SMT, cleaning up by trying to detect and remove incorrect alignments can actually degrade performance. Our experimental results confirm their findings on some datasets. Based on our method, sometimes filtering noisy data leads to unexpected results. The reason is two-fold: on the one hand, the non-literal parallel data makes false positive in noisy data detection; on the other hand, large-scale SMT systems is relatively robust and tolerant to noisy data, especially when we remove frequency-1 phrase pairs. Therefore, we propose to integrate the importance scores when re-estimating phrase pair probabilities in this paper. The importance scores can be considered as a kind of contribution constraint, thereby high-quality parallel data contributes more while noisy parallel data contributes less.

4 Conclusion and Future Work

In this paper, we develop an effective approach to clean the bilingual data using graph-based random walk. Significant improvements on several datasets are achieved in our experiments. For future work, we will extend our method to explore the relationships of sentence-to-sentence and phrase-to-phrase, which is beyond the existing sentence-to-phrase mutual reinforcement.

Acknowledgments

We are especially grateful to Yajuan Duan, Hong Sun, Nan Yang and Xilun Chen for the helpful discussions. We also thank the anonymous reviewers for their insightful comments.

References

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117.

Jeffrey Dean and Sanjay Ghemawat. 2008. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61, Sydney, Australia, July. Association for Computational Linguistics.

Cyril Goutte, Marine Carpuat, and George Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *Proceedings of AMTA 2012*, San Diego, California, October. Association for Machine Translation in the Americas.

Long Jiang, Shiquan Yang, Ming Zhou, Xiaohua Liu, and Qingsheng Zhu. 2009. Mining bilingual data from the web with adaptively learnt patterns. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 870–878, Suntec, Singapore, August. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003 Main Papers*, pages 48–54, Edmonton, May-June. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Philip Resnik and Noah A Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

Lei Shi, Cheng Niu, Ming Zhou, and Jianfeng Gao. 2006. A dom tree alignment model for mining parallel data from the web. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 489–496, Sydney, Australia, July. Association for Computational Linguistics.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 552–559, Prague, Czech Republic, June. Association for Computational Linguistics.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Joern Wuebker, Arne Mauser, and Hermann Ney. 2010. Training phrase translation models with leaving-one-out. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 475–484, Uppsala, Sweden, July. Association for Computational Linguistics.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528, Sydney, Australia, July. Association for Computational Linguistics.