Document-level Consistency Verification in Machine Translation

Tong Xiao[†‡], Jingbo Zhu[†‡], Shujie Yao[†], Hao Zhang[†‡]

[†]Natural Language Processing Lab, Northeastern University
[‡]Key Laboratory of Medical Image Computing, Ministry of Education
{xiaotong,zhujingbo}@mail.neu.edu.cn
{yaosj, zhanghao}@ics.neu.edu.cn

Abstract

Translation consistency is an important issue in document-level translation. However, the consistency in Machine Translation (MT) output is generally overlooked in most MT systems due to the lack of the use of document contexts. To address this issue, we present a simple and effective approach that incorporates document contexts into an existing Statistical Machine Translation (SMT) system for document-level translation. Experimental results show that our approach effectively reduces the errors caused by inconsistent translations (25% error reduction). More interestingly, it is observed that as a "bonus" our approach is able to improve the BLEU score of the SMT system.

1 Introduction

Recently, document translation (or document-level translation) has received a growing interest in the fields of both human translation and Machine Translation (MT). For example, with the increasing demand in cross-lingual patent retrieval and filing patent applications in foreign countries, patent document translation has been recognized as one of the fundamental issues in patent processing and related applications (Fujii et al., 2008).

Unlike traditional sentence-level translation, document-level translation requires translators to hold a global view of the whole document rather than to focus on translating each source sentence individually. There are a number of critical issues in document-level translation (Catford, 1965). One of them is the issue of *translation consistency* (Nida, 1964). E.g. when we translate a term within a document, we prefer to keep the same translation throughout the whole document no matter how many times it is repeated. This is especially important in certain applications such as translation of legal documents and government documents. In some cases, consistency is even regarded as one of the primary quality measurements of translation (He et al., 2009).

However, directly modeling the translation problem on the whole document is a challenging issue due to its high complexity. It is even intractable to implement or run such a translation system in practice. To ease the problem, a general solution is to view the source text as a series of independent sentences and do translation using sentence-level SMT approaches. However, in this case, document contexts – essential factors to document-level translation – are generally overlooked either in training or inference (i.e. decoding in SMT) stage.

In this paper, we address the issue of how to introduce document contexts into current SMT systems for document-level translation. In particular, we focus on translation consistency which is one of the most important issues in document-level MT. We propose a 3-step approach to incorporating document contexts into a traditional SMT system, and demonstrate that our approach can effectively reduce the errors caused by inconsistent translation. More interestingly, it is observed that using document contexts is promising for BLEU improvement.

2 Related Work

To date, only a few studies have improved MT systems with the use of document contexts. For example, Brown (2008) proposed a method to improve SMT and Example-Based Machine Translation (EBMT) systems using document-level similarity between the documents in the training corpus and the input document. Another example is (Zhao and Xing, 2007) in which a bilingual topic model was proposed to capture the document-level topical aspects of SMT. However, no previous work has addressed the issue of translation consistency in document-level MT.

The problem discussed in this paper is similar to the lexical selection problem in SMT (Wu and Palmer, 1994). There have been some attempts at using context-dependent features to select appropriate target lexical items for SMT systems (Carpuat and Wu, 2005; Carpuat and Wu, 2007; Chan et al., 2007). However, these studies were all in the scenario of sentence-level MT. By contrast, we focus more on using document contexts to address the issue in document translation. Actually, the translation consistency issue has been discussed in some related tasks. For example, Wang et al. (2007)'s work showed that consistency information was very helpful in dealing with the out-of-vocabulary (OOV) problem for Chinese word segmentation.

3 Document-level Consistency Verification

Given a source document D_f, the task of document-level SMT is to find an optimal target document D_e^* by:

$$D_e^* = \arg\max_{D_e} \Pr(D_e \mid D_f) \qquad (1)$$

Since modeling $\Pr(D_e \mid D_f)$ is not an easy task, the problem is generally further decomposed into a group of sub-problems. Supposing that D_f consists of a sequence of source sentences $\{f_1, ..., f_n\}$, we can find the optimal translation e_i^* for each f_i by:

$$e_i^* = \arg\max_{e_i} \Pr(e_i \mid f_i) \qquad (2)$$

It is relatively easy to solve Equation (2) using sentence-level SMT approaches. Thus, the final translation of D_f is generated by gluing the sequence of sentence translations, that is $D_e^* = \{e_1^*, ..., e_n^*\}$.

In general this method is regarded as a *baseline* method to implement document-level translation with existing sentence-level SMT systems. In this work, we extend this method to address the issue of translation consistency using document contexts. We first obtain an initial translation result D_e^* using the baseline method, and then perform the following three steps to improve the translation consistency of the baseline method.

***Step* 1**. First, we identify the *ambiguous words* W in D_f whose translations are inconsistent in D_e^*.

***Step* 2**. We then obtain a set of consistent translations $C(w)$ for each $w \in W$ according to the distribution of w's translation over the target document.

***Step* 3**. Based on the results of Step 1 and Step 2, we generate a new translation D_e^{**} for D_f, guaranteeing that the words in W have consistent translations in D_e^{**}.

In the following parts of this section, we describe them in detail.

Step 1: Identification of Ambiguous Words

In document translation, a source word may have more than one different translation. Here we say that a source word has *inconsistent translations* if it has two or more different translations over a document. Further we define *ambiguous words* to be the source words that have inconsistent translations in the output of the baseline system. The identification of ambiguous words is simple. We define that a source word w is an ambiguous word if and only if it satisfies the following constraints.

1). w appears more than 2 times in D_f

2). w is a term[1]

3). w has multiple *inequivalent* translations in D_e^*

As we focus more on verifying the consistency in translations of content words (e.g. nouns), we further define that two translations of w are *inequivalent* if they are not the same string after stemming and removing functional words. For example, "railway" and "railways" are equivalent, while "day" and "festival" are inequivalent.

Step 2: Obtaining Consistent Translations

For each ambiguous word w, we obtain a set of translation candidates $C(w)$ which will then be used to generate consistent translations for w in the final step. Let $O(w) = \{t_1, ..., t_n\}$ be the set of candidate translations for w, and $F(t)$ be the frequency of t occurring in the target document. $C(w)$ is built from $O(w)$ by selecting $t_i \in O(w)$ that appears most

[1] We use a term base (consisting of about 500K Chinese named entities downloaded from web) and a rule-based system for term identification in this study.

frequently (i.e. with the highest $F(t_i)$) in the outputs of our baseline system[2]. Figure 1 shows an example. In Step 1, we obtain an ambiguous word 对象 which has two translation candidates *object* and *girl friend* (i.e. O(对象) = {*object*, *girl friend*}). In the initial translation result, *object* appears more frequently than *girl friend* (i.e. F(*object*) > F(*girl friend*)). Therefore, *object* is selected as the translation for the ambiguous word 对象 in translating D_f (i.e. C(对象) = {*object*}).

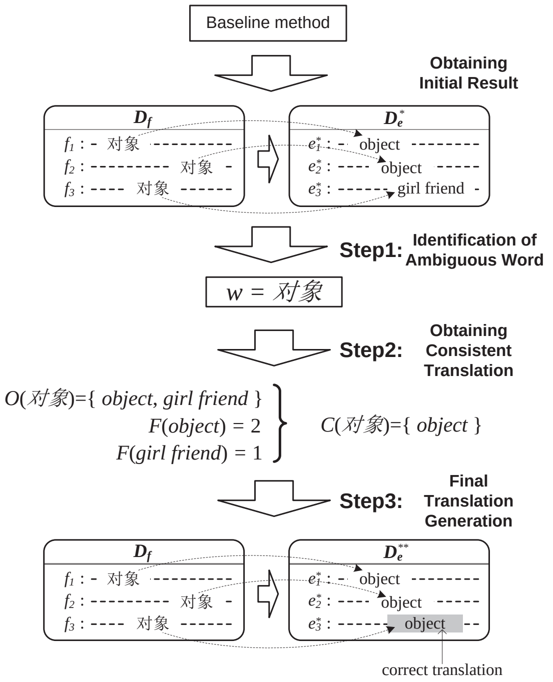
Baseline method
Obtaining Initial Result
D_f
f_1 : - 对象 ---------
f_2 : --------- 对象 -
f_3 : ---- 对象 ------
D_e^*
e_1^* : - object --------
e_2^* : ---- object -----
e_3^* : ------ girl friend -
Step1: Identification of Ambiguous Word
w = 对象
Step2: Obtaining Consistent Translation
O(对象)={ object, girl friend }
F(object) = 2
F(girl friend) = 1
C(对象)={ object }
Step3: Final Translation Generation
D_f
f_1 : - 对象 ---------
f_2 : --------- 对象 -
f_3 : ---- 对象 ------
D_e^{**}
e_1^* : - object --------
e_2^* : ---- object -----
e_3^* : ------ object --
correct translation

Figure 1: An illustration of our approach

Here $O(w)$ and $F(t)$ model the document contexts used in obtaining the consistent translations for w. In this work, $O(w)$ is calculated by collecting all the translations of w in the k-best translations for each f_i.

$$O(w) = \bigcup_{f_i \in D_f} \bigcup_{1 \le j \le k} \bigcup_{\substack{\forall t: w \to t, \\ \delta(t, e_{ij}) = 1}} \{t\} \qquad (3)$$

where e_{ij} denotes the j-th best translation of f_i, $w \rightarrow t$ denotes that t is a translation of w, and $\delta(t, e_{ij})$ denotes an indicator function that returns 1 if t appears in e_{ij}[3], otherwise 0. k is a parameter that controls the scope of obtaining the candidate translations of w, e.g. a larger k means that more translations are taken into account in calculating $O(w)$.

To estimate $F(t)$, a straightforward solution is to count the occurrence of t in the k-best translations of each f_i, that is, each occurrence of t has a count of unit one. However, the k-best translations are not simply a set of translations. Instead, it is typically viewed as a weighted list in which each translation has a weight (or probability) indicating the "confidence" that the translation model has on it. So a more reasonable way is to penalize e_{ij} according to its weight when counting the occurrence of t. Motivated by this idea, we use *fractional count* (*fcount*) instead of unit one to count the occurrence of t. The fractional count of t in e_{ij} for the i-th source sentence is defined to be:

$$fcount(t, f_i, e_{ij}) = \Pr(e_{ij} \mid f_i) \cdot \delta(t, e_{ij}) \qquad (4)$$

where $\Pr(e_{ij} \mid f_i)$ is the posterior probability of e_{ij} for the given source sentence f_i. In this work, the translation posterior $\Pr(e_{ij} \mid f_i)$ is computed in a log-linear fashion.

$$\Pr(e_{ij} \mid f_i) = \frac{\exp(\alpha \cdot Score(e_{ij}, f_i))}{\sum_{j'=1}^{k} \exp(\alpha \cdot Score(e_{ij'}, f_i))} \qquad (5)$$

where $Score(e_{ij}, f_i)$ is the model score determined by the SMT model, and α is a scaling factor that determines how flat or peaked the distribution is.

Finally, $F(t)$ is calculated using the fractional counts of t over the target document. Two methods can be used to obtain $F(t)$. The first method (M1) accumulates the fractional counts over all the k-best lists by:

$$F(t) = \sum_{f_i \in D_f} \sum_{j=1}^{k} fcount(t, f_i, e_{ij}) \qquad (6)$$

Alternatively, the second method (M2) only considers the dominant count for each individual source sentence (Equation 7).

$$F(t) = \sum_{f_i \in D_f} \max_{1 \le j \le k} \{ fcount(t, f_i, e_{ij}) \} \qquad (7)$$

After we process all the sentences in D_f, $C(w)$ is generated using $O(w)$ and $F(t)$.

[2] When more than one translation has the highest frequency, all of them are selected.

[3] t can be viewed as a sub-string of e_{ij} when $\delta(t, e_{ij}) = 1$.

Step 3: Final Translation Generation

Finally, we incorporate $C(w)$ into the baseline system to generate consistent translations of w over the target document. A straightforward method is ***post-editing***. Suppose that t is a translation of w in the initial result D_e^*. If $t \notin C(w)$, we replace t with a translation in $C(w)$[4]; otherwise, keep t unchanged. This method guarantees that all the translations of w are consistent with $C(w)$ in the target document.

Although post-editing is simple to improve the translation consistency of the baseline system, it generally results in more disfluencies in outputs due to lack of the use of local contexts in translation. To address this problem, we can choose another solution called ***re-decoding*** (or multi-pass decoding). In re-decoding, the translation options of w is first filtered using $C(w)$. If a translation option of w is inequivalent to any $t \in C(w)$, it is removed from the translation table. We then decode the source sentences again to generate new translations using the filtered table. Compared to post-editing, this method can generate more smooth translations using the SMT model.

Figure 1 illustrates our approach with a tiny example. After all three steps, the translation of *对象* focuses on *object* which is the correct translation for *对象* for the document.

4 Experiments

4.1 Baseline System

Our experiments were conducted on Chinese-English translation based on the open-source phrase-based MT system *NiuTrans*[5]. The *NiuTrans* uses two reordering models, including a maximum entropy-based lexicalized reordering model (Xiong et al., 2006) and a MSD reordering model (Koehn et al., 2007). In addition, it adopts all standard features used in the state-of-the-art SMT system Moses (Koehn et al., 2007), such as bi-directional phrase translation probabilities and n-gram language model. The feature weights were optimized using MERT (Och, 2003). By default, the distortion limit was set to 8, k was set to 1 (Equations 3, 5-7), and α was set to 0 (Equation 5).

4.2 Data Sets and Evaluation Methods

The phrase translation table and reordering model were extracted from a corpus of about 370K bilingual sentences[6]. A 5-gram language model was built on the Xinhua portion of the English Gigaword corpus in addition to the English side of the bilingual corpus. For development and test data, we used NIST Chinese-to-English evaluation sets of MT03 (99 documents, 919 sentences) and MT05 (97 documents, 1082 sentences), respectively. Both of the data sets were from news reports where the translation consistency is required.

We manually annotated a number of checkpoints to check whether ambiguous words were correctly translated. The checkpoints were obtained in two steps: 1) we first selected ambiguous words[7] as the candidate checkpoints; 2) and then manually removed the candidate checkpoints where the consistency of translation is not strongly required. The system was evaluated by the number of errors at checkpoints. We also reported the BLEU(-SBP) (Chiang et al., 2008) score to show the impact of our approach on translation accuracy.

4.3 Results in Default Settings

We first investigate the effectiveness of our methods on error reduction in the default settings. Table 1 compares various methods in terms of the number of errors at checkpoints, where *Post* and *Rede* stand for the post-editing and the re-decoding methods used in final translation generation (Step 3) respectively, M1 and M2 stand for the two counting methods shown in Equations (6-7). We see that all our proposed methods are effective to reduce the incorrect translations at checkpoints. In most cases, they lead to over 25% error reduction. From this table we also observe that, post-editing-based final translation generation is more effective in error reduction than re-decoding-based final translation generation. This can be explained by the fact that although the re-decoding-based method is good at generating appropriate boundary words for phrase translation, it weakens the constraint of translation consistency, and thus is more likely to lose "consistency" in final outputs.

[4] If $|C(w)| > 1$, we randomly select one element from $C(w)$.

[5] http://www.nlplab.com/NiuPlan/NiuTrans.html

[6] LDC Category: LDC2005T10, LDC2003E07, LDC2003E14 and LDC2005T06

[7] If w occurs m times in a document, there will be m checkpoints for w in this document.

Method	# of errors (Dev)	# of errors (Test)
Baseline	268	333
Post + M1	187	244
Post + M2	183	253
Rede + M1	190	245
Rede + M2	193	247

Table 1: Comparison of various methods in terms of the number of errors at checkpoints.

We then study the impact of our methods on BLEU scores. Table 2 shows the BLEU scores of various different methods, where column *Check* means the BLEU scores on the set of sentences containing at least one checkpoint, and column *Full* means the BLEU scores on the full evaluation sets of NIST MT03 and MT05. As shown in Table 2, post-editing-based final translation generation results in a lower BLEU score compared to the baseline method. A possible reason for this phenomenon is that when post-editing corrects certain types of translation error, it in turn brings new errors into the translations. For example, in some cases, the post-editing method leads to incorrect preposition collocations despite a correct translation of content word. Compared to the post-editing method, the re-decoding method achieves higher BLEU scores because it can select more appropriate translations for ambiguous words and their local contexts using the translation model and language model, rather than editing the final translation naively. Also, re-decoding can stably outperform the baseline method, in some cases even achieve a +0.4 BLEU improvement.

Method	BLEU4[%] (Dev)		BLEU4[%] (Test)	
	Check	Full	Check	Full
Baseline	35.99	35.62	33.12	33.69
Post + M1	35.68	35.47	32.96	33.63
Post + M2	35.62	35.45	32.79	33.56
Rede + M1	36.09	35.67	33.50	33.87
Rede + M2	36.01	35.63	33.41	33.83

Table 2: Comparison of various methods in terms of BLEU(-SBP) scores.

4.4 Impact of *k*-best List Size

Next, we investigate the impact of *k* on error reduction and BLEU improvement (Figures 2-7). Figures 2-3 show the numbers of errors at different settings of *k*. We see that enlarging *k* is very effective to reduce the error further. When $k \geq 5$ our methods generally achieve 35% error reduction, which is much higher than that of $k = 1$ (25% error reduction). Also as shown in Figures 2-3, the post-editing method is relatively more effective in generating final translations with fewer errors. In most cases, it outperforms its counterpart (i.e. re-decoding method) over 5% fewer errors.

However, the trends shown in Figures 2-3 are not held in Figures 4-7. The BLEU scores do not vary too much when we adjust the parameter *k*. At each setting of *k*, as expected, the re-decoding method outperforms the post-editing method due to more smooth translations selected by the language model. However the BLEU improvement or decreasement is not significant.

In addition, it seems that the parameter *k* affects the results of obtaining consistent translations more. Compared to the M1 method (Equation 7), the M2 method (Equation 6) benefits more from a larger *k*. When the M2 method is utilized, the BLEU score can be further improved over 0.15 points when *k* is set to 5, which in turn results in an overall improvement of 0.5 BLEU-points over the baseline on the data sets consisting of the sentences with checkpoints (Figures 4-5).

4.5 Impact of α

We also study how the distribution of posterior probability affects the translation results. As the systems perform better at $k \geq 5$ in the previous experiments, we set *k* to 5. Figures 8-9 show that further error reduction is achieved on both the development and test sets when α is between 0.005 and 0.1 (inclusive). It indicates that our methods can make benefits more from a relatively unsmoothed distribution rather than a uniformed distribution. However, on the other side, a too skewed distribution ($\alpha > 0.1$) generally results in more errors.

More interestingly, we observe that the distribution of posterior probability has a minor influence on BLEU scores (Figures 10-13). In most cases, there is a less than 0.1 BLEU point volatile. This result confirms the fact that the error reduction does not always lead to the BLEU improvements due to the different views adopted in defining translation error and BLEU score.

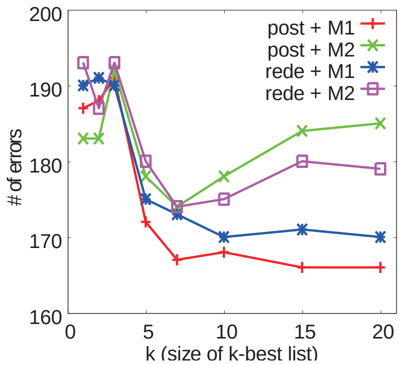

Figure 2: Error number against *k*
(full Dev set)

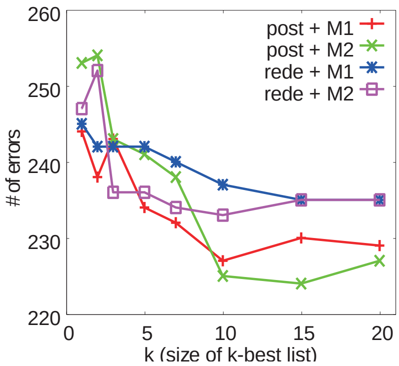

Figure 3: Error number against *k*
(full Test set)

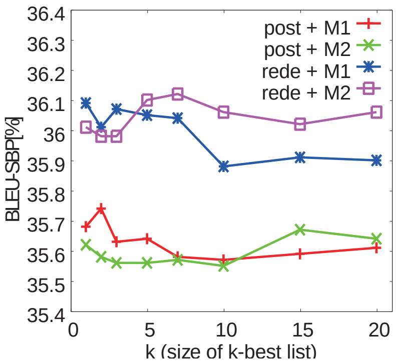

Figure 4: BLEU-SBP against *k*
(Dev sentences with checkpoints)

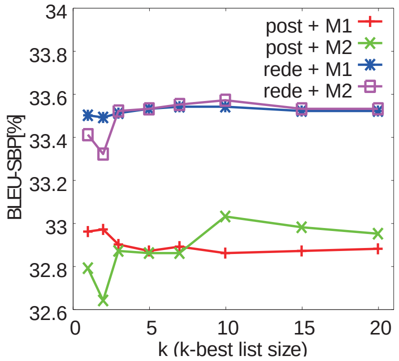

Figure 5: BLEU-SBP against *k*
(Test sentences with checkpoints)

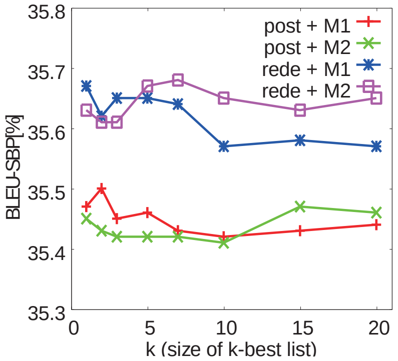

Figure 6: BLEU-SBP against *k*
(full Dev set)

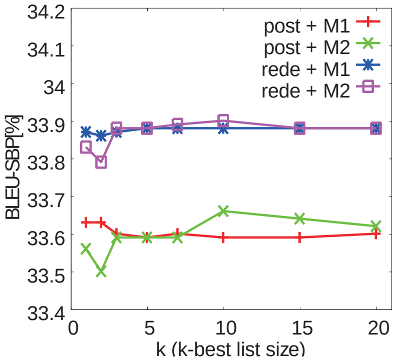

Figure 7: BLEU-SBP against *k*
(full Test set)

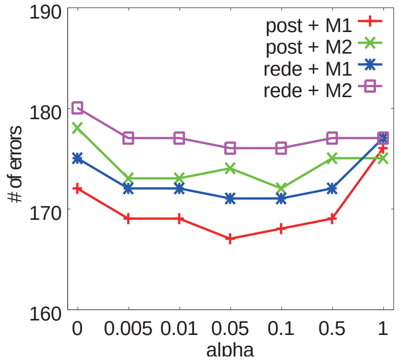

Figure 8: Error number against α
(full Dev set)

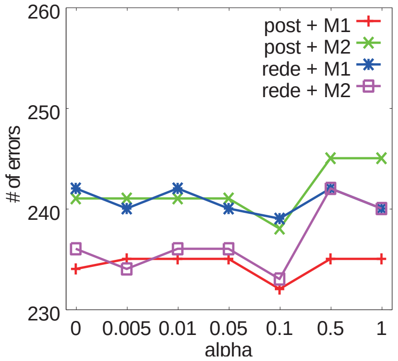

Figure 9: Error number against α
(full Test set)

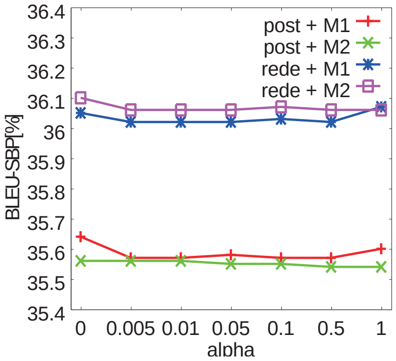

Figure 10: BLEU-SBP against α
(Dev sentences with checkpoints)

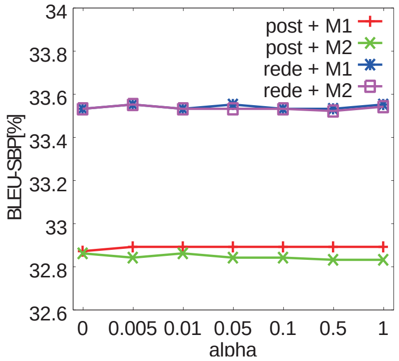

Figure 11: BLEU-SBP against α
(Test sentences with checkpoints)

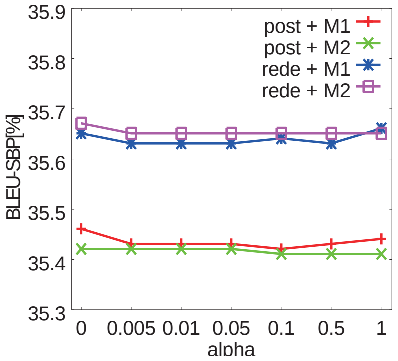

Figure 12: BLEU-SBP against α
(full Dev set)

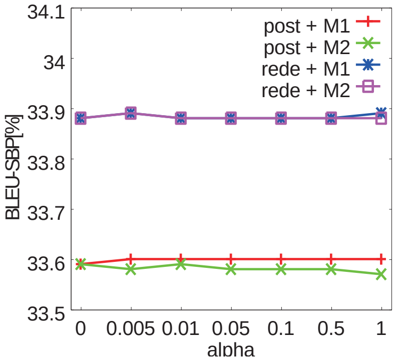

Figure 13: BLEU-SBP against α
(full Test set)

5 Analysis

We analyze the data to find what problems remain after our improvement. Table 3 shows the numbers of correct and wrong operations performed at checkpoints. Here *correct operation* refers to the operation that correctly revises the incorrect translation of ambiguous word (i.e. inconsistent and incorrect translation over the document), and *wrong operation* refers to the operation that incorrectly revises the translation (i.e. inconsistent but correct translation). We see that our methods make over 80% correct operations in most cases. Further, we find that there are three major problems with our methods.

Method	Dev		Test	
	correct	wrong	correct	wrong
Post + M1	91	10	116	27
Post + M2	100	15	112	32
Rede + M1	123	19	134	25
Rede + M2	107	30	131	27

Table 3: Comparison of various methods in terms of the number of correct/wrong operations.

Problem 1. The correct translation is not contained in the set of translation candidates *C*(*w*). Almost all wrong operations are caused by this problem. The reason lies in that the baseline system does not generate any initial translations that contain the correct translation for ambiguous words. For example, when our baseline systems translates *反分裂* in a document, it does not find any correct translations for *反分裂*. In this case, it is very hard to improve the translation of *反分裂* from the initial result. We further find that the problem is mainly due to the limited coverage of our translation table, that is, the table does not contain the correct translation options for *反分裂*. To alleviate this problem, a possible solution might be that we use more bilingual data to obtain a larger translation table, and thus improve the initial results generated by the baseline system.

Problem 2. *F*(*t*) does not provide us with enough information to make correct decisions. For example, in Figure 14, *计划* has two different translations *plan* and *program* in the initial translation result, while the reference translation focuses on *program* only. However, since *F*(*plan*) = *F*(*program*), we cannot make a decision that whether to select *plan* or *program* as the consistent translation we should focus on. Consequently, our method does not change the initial result. We find that over 70% of the remaining errors (Table 1) are due to this problem. Instead of using only word frequency, more sophisticated features are required for further improvement.

Problem 3. Translation consistency is not required in some cases. We find that over 30% of the words processed by our method do not need consistent translation over the document. For example, *区域* in general has two translations *area(s)* and *region(s)*. Both of them are acceptable, and may appear respectively in the two reference translations. However, our method forces to translate *区域* into *areas* due to its high frequency over the document. Although this method does not have the error of inconsistent translation, it leads to disfluencies and a lower BLEU score due to the mismatching between the MT output and the reference translation. One of the possible reasons for this problem is that we simply define the inconsistent translation upon certain groups of words (e.g. terms), which suggests an interesting direction: studying in which cases MT systems need the consistency in document-level translation.

6 Conclusion

We have presented results showing that using document contexts is promising for use in document-level machine translation. When working with a state-of-the-art SMT system, we were able to reduce over 25% errors caused by inconsistent translations over a document. Of more interesting is that as a "bonus" our method was able to improve the translation accuracy of the MT system.

Our work is an exploration of an interesting issue concerned by MT researchers. Though a little primitive, it shows promising results and encourages us to go on the study on this topic. For example, we will extend our focus from ambiguous words to ambiguous phrases, as phrase (or *n*-gram) is the basic translation unit in most state-of-the-art SMT systems. In addition, other issues, such as abbreviation translation, are also important for document-level translation, and worth studying in our future work.

Baseline (Initial result)	This work (Post + M1 in default settings)	Reference
General of Japan today began to promote an advanced network ***plan***, hoping that by 2010, 20% of labor from japan could "commute" through ***program***, and reduce the pressure brought about by the office and family problems.	General of Japan today began to promote an advanced network ***plan***, hoping that by 2010, 20% of labor from Japan could "commute" through ***program***, and reduce the pressure brought about by the office and family problems.	Japan's ministry of internal affairs and communications today launched an advanced internet ***program***, hoping that 20% of the Japanese workforce could "telecommute" through the ***program*** before 2010 to reduce pressures from office life and resolve the problem of family splits.
The province will the general reaction, and expand the scope of such advanced ***program*** until 20% of all the 2,500 workers can work at home in the before 2006.	The province will the general reaction, and expand the scope of such advanced ***program*** until 20% of all the 2,500 workers can work at home in the before 2006.	The ministry of internal affairs and communications will expand the scope of the ***program*** based on feedbacks to increase the number of telecommuting employees working at home to 20% of its 2,500 strong workforce.
The general provincial official said that the employees can chat through the internet and electronic meeting a ***plan*** that can "increase the efficiency of the labor market".	The general provincial official said that the employees can chat through the internet and electronic meeting a ***plan*** that can "increase the efficiency of the labor market".	A ministry official says the ***program***, which enables employees to use chat rooms and teleconferencing, is expected to "improve workplace efficiency".

Figure 14: Example outputs. The translations of ambiguous words *计划* are bolded, colored and italicized

Acknowledgements

This work was supported in part by the National Science Foundation of China (60873091; 61073140), Specialized Research Fund for the Doctoral Program of Higher Education (20100042110031), and the Fundamental Research Funds for the Central Universities.

References

Ralf D. Brown. 2008. Exploiting Document-Level Context for Data-Driven Machine Translation. In *Proc. of* AMTA 2008, pages 46-55.

Marine Carpuat and Dekai Wu. 2005. Word sense disambiguation vs. statistical machine translation. In *Proc. of* ACL 2005, pages 61-72.

Marine Carpuat and Dekai Wu. 2007. How phrase sense disambiguation outperforms word sense disambiguation for statistical machine translation. In *Proc. of* TMI 2007, pages 43-52.

J. C. Catford. 1965. *A Linguistic Theory of Translation: An Essay in Applied Linguistics*, Oxford University Press, London.

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, Steve DeNeefe, Yee Seng Chan and Hwee Tou Ng. 2008. Decomposability of translation metrics for improved evaluation and efficient algorithms. In *Proc. of* EMNLP 2008, pages 610–619.

Atsuchi Fujii, Maso Utiyama, Mikio Yamamoto and Takehito Utsuro. 2008. Overview of the Patent Translation Task at the NTCIR-7 Workshop. In *Proc. of* the 7th NTCIR Workshop Meeting, pages 389-399.

Xiaodong He, Masaki Itagaki and Takako Aikawa. 2009. Validation of the consistency of automatic terminology translation. Patent application, Microsoft Corporation.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of* ACL 2007, Demonstration Session.

Eugene A. Nida. 1964. *Toward a Science of Translating*. Brill Academic Publishers.

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of* ACL 2003, Japan, pages 160-167.

Zhenxing Wang, Changning Huang and Jingbo Zhu. 2007. The Character-based CRF Segmenter of MSRA&NEU for the 4th Bakeoff. In *Proc. of* the Sixth SIGHAN Workshop on Chinese Language Processing, pages 98-101.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *Proc. of* ACL 1994, pages 133-138.

Deyi Xiong, Qun Liu and Shouxun Lin. 2006. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In *Proc. of* ACL 2006, Sydney, pages 521-528.

Bing Zhao and Eric P. Xing. 2007. HM-BiTAM: Bilingual Topic Exploration,Word Alignment, and Translation. In *Proc. of* NIPS 2007.