Document-level Re-ranking with Soft Lexical and Semantic Features for Statistical Machine Translation

Chenchen Ding tei@mibel.cs.tsukuba.ac.jp
Department of Computer Science, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, Japan
Masao Utiyama mutiyama@nict.go.jp
Eiichiro Sumita eiichiro.sumita@nict.go.jp
National Institute of Information and Communications Technology
3-5 Hikaridai, Seikacho, Sorakugun, Kyoto, 619-0289, Japan

Abstract

We introduce two document-level features to polish baseline sentence-level translations generated by a state-of-the-art statistical machine translation (SMT) system. One feature uses the word-embedding technique to model the relation between a sentence and its context on the target side; the other feature is a crisp document-level token-type ratio of target-side translations for source-side words to model the lexical consistency in translation. The weights of introduced features are tuned to optimize the sentence- and document-level metrics simultaneously on the basis of *Pareto optimality*. Experimental results on two different schemes with different corpora illustrate that the proposed approach can efficiently and stably integrate document-level information into a sentence-level SMT system. The best improvements were approximately 0.5 BLEU on test sets with statistical significance.

1 Introduction

State-of-the-art statistical machine translation (SMT) systems (Koehn et al., 2007) have achieved good performance for many translations, such as French-to-English translation. The success can be attributed to the statistical model used in translation and the huge data for model training. However, the well-developed techniques of SMT are mainly focused on the sentence-level translation, i.e., the models are trained on the parallel corpus of sentence pairs, and the translation is conducted sentence-by-sentence. Because in practice sentences are usually contained in a document and surrounded by context, recent research has begun to focus on enhancing SMT systems with the addition of document-level information.

As to the features of document-level translation, a frequently discussed issue is lexical consistency in translation: i.e., words tend to be translated consistently in a document (Carpuat, 2009; Carpuat and Simard, 2012). There are also detailed discussions around the consistency of different parts of speech (Guillou, 2013; Meyer and Webber, 2013). On the basis of lexical consistency theory, many researchers focus on increasing the lexical consistency in translation (Tiedemann, 2010; Xiao et al., 2011; Ture et al., 2012). Beyond lexical consistency, there are attempts at using lexical cohesion in translation (Ben et al., 2013; Xiong et al., 2013a,b), which considers the semantic relation between words. Rather than the lexical features, the topic of

documents is also taken as a feature in some recent research (Gong et al., 2011; Eidelman et al., 2012; Xiong and Zhang, 2013; Hasler et al., 2014).

Among the different features, lexical consistency is the simplest feature because it only considers the lexical words themselves. In contrast, lexical cohesion involves more semantic information, such as *hypernyms* and *hyponyms*, usually requiring a word-net. Approaches that use the document topic as a feature usually require training data, such as a document-level parallel corpus, in the training or decoding phases.

In this paper, we propose an approach that considers both the lexical consistency and semantic relation on the document level. The approach first uses an off-the-shelf SMT system to conduct the sentence-level translation, where both the training and decoding are on the sentence level. Then we introduce two document-level features, one using the word-embedding technique to model the semantic relation of context on the target side and the other using a token-type ratio to model the consistency in translation. With the two document-level features, we conduct a further decoding on the document level to get a better combination of sentence-level translation within a document. As to the weights of the introduced features, we utilize a multi-objective learning approach based on the *Pareto optimality* (Duh et al., 2012) to simultaneously optimize the sentence-level and document-level metrics. The proposed approach requires no word-net or document-level parallel corpus for model training. Instead, it requires a vector list of the target-side vocabulary by word embedding and a small development set of parallel document pairs to tune the weights of document-level features.

The remainder of the paper is organized as follows. In Section 2, we mention the related work around using document-level information in translation. In Section 3, we describe our proposed approach. Section 4 presents experimental results, and Section 5 is the discussion, where we compare the proposed approach with a *consistency verification* approach (Xiao et al., 2011). Section 6 contains the conclusions and future work.

2 Related Work

For the approaches focusing on the lexical consistency, an early attempt is the work of Tiedemann (2010), where decaying cache models for both language and translation models are used for SMT. The cache models give the SMT system a preference for recently used words and translation rules. The approach succeeded for an out-of-domain test set but failed for an in-domain test set. Tiedemann (2010) mentions that the cache model may be "*risky*". In Xiao et al. (2011), a re-decoding approach for a baseline SMT system is proposed to ensure lexical consistency in translation, with quite detailed manual analysis of the experiment results. The approach also has an improved BLEU score, which the authors mention as a *bonus*. Ture et al. (2012) used a force-decoding approach for an SCFG-based translation system with several *Okapi BM25* term weights. These works are based on the "*one translation per corpus*" constraint discussed in Carpuat (2009). On the other hand, the report in Carpuat and Simard (2012) asserted "*SMT translates document remarkably consistently, even without document knowledges*." In our opinion, this is a complex issue that may depend on the data used or even the language pair in the translation task.

As to the approaches using lexical cohesion, Ben et al. (2013) and Xiong et al. (2013a) use semantic relations in a word-net to identify bilingual *hypernym* and *hyponym* relations in translation. In Xiong et al. (2013b), a thesaurus is used to construct the source-side lexical chain. These approaches step forward into the field of the semantic; thus, they require the help of particular linguistic resources.

Document-level topic-based approaches also exist (Gong et al., 2011; Eidelman et al., 2012; Xiong and Zhang, 2013; Hasler et al., 2014), which introduce extra topic models into the translation process to improve the word selection for specific topics. Usually, the topic model

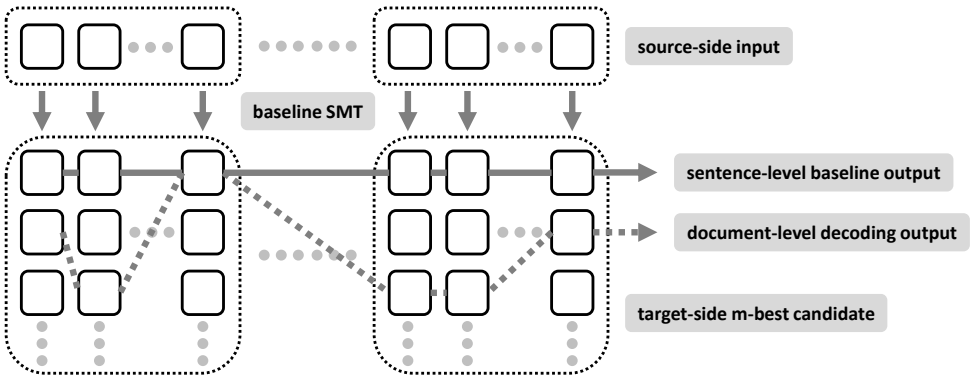

Figure 1: *Overall process of proposed approach. Sentences are represented by boxes with bold frame, and documents are represented by boxes with fine-dashed frame. A baseline SMT system on sentence level will take the set of* 1*-best translations of each input sentence as the final output (marked by the solid horizontal arrow). The proposed approach conducts a document-level decoding on the target-side* m*-best candidate lists of source-side sentences to find a better combination (marked by the dashed zigzag arrow).*

is statistical and needs to be trained on monolingual or bilingual document-level data. Along with the feature of lexical cohesion, the topic is a sophisticated feature that must be supported by extra resources.

Many approaches using document-level features require to modify the decoder of a baseline system to adapt to their features in decoding. Research mainly focusing on the decoding and tuning algorithm, such as the series work of Hardmeier et al. (2012) and Stymne et al. (2013), extends the traditional sentence-based SMT system to be able to collaborate with document-level features.

As to our approach, the features used can model the lexical consistency as well as semantic relation at a certain level while not being as rigid as the features/operations on the very lexical level that many previous approaches use. We assume that these features, combined with the multi-objective tuning, will provide a robust and stable way to take advantage of document-level information in an SMT system.

3 Proposed Approach

3.1 Overview

The proposed approach is essentially a re-ranking process in a document-level decoding (Fig. 1). We first use an off-the-shelf baseline SMT system to translate a document sentence-by-sentence, obtaining the m-best translation candidates for each sentence. The baseline SMT system can be trained and tuned in a standard way with sentence-level parallel data. Then, we conduct a decoding on the document level to find *good* combinations among the m-best candidate sentences. The search is realized in a cube-pruning way (Chiang, 2007). Here, we use *good* to mean that the combinations are good for both sentence- and document-level metrics under the *Pareto optimality* (Duh et al., 2012). As far as we know, this is the first attempt to apply document-level re-ranking in an SMT system.

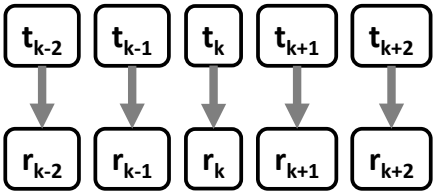

Figure 2: *In S_{eval} (BLEU)*, every translation t will be compared with its reference r.

3.2 Notation

In the following description, we use D to denote an input document on the source-side language composed of n sentences, which are $\{s_1, s_2, \cdots s_n\}$. The reference translation of D on the target-side language is denoted by D^r, composed of $\{r_1, r_2, \cdots r_n\}$, where sentence r_k $(1 \leq k \leq n)$ is the reference translation of s_k. Each sentence s_k $(1 \leq k \leq n)$ in D has an m-best translation candidate composed of $\{t_k^1, t_k^2, \cdots, t_k^m\}$. Over the total n m-best candidate lists, we search for a combination $C = \{t_1^{c_1}, t_2^{c_2}, \cdots t_n^{c_n}\}$ $(1 \leq c_k \leq m, 1 \leq k \leq n)$ for optimization. Generally, a test set contains multiple l-documents of $\{D_1, D_2, \cdots D_l\}$; hence, correspondingly we search $\{C_1, C_2, \cdots C_l\}$.

3.3 Optimization Function

For optimization, we use two objective metrics: a sentence-level metric (S_{eval}) and a document-level metric (D_{eval}), as follows.

$$S_{eval}(\{C_1, \cdots, C_l\}, \{D_1^r, \cdots, D_l^r\}) = \text{BLEU}(\{C_1, \cdots, C_l\}, \{D_1^r \cdots, D_l^r\}) \tag{1}$$

Using the BLEU score for the S_{eval}, a candidate translation will be evaluated with its reference translation, sentence by sentence, disregarding document-level information (Fig. 2).

$$D_{eval}(\{C_1, \cdots, C_l\}, \{D_1^r, \cdots, D_l^r\}) = \underset{1 \leq i \leq l}{\text{avg}} \{\underset{k}{\text{avg}}\, \text{diff}(\text{context of } t_k \in C_i, r_k \in D_i^r)\} \quad (1 \leq k \leq length(C_i)) \tag{2}$$

For the D_{eval}, we evaluate the context of a candidate translation with its reference translation (Fig. 3). In Exp. (2), avg represents average and the function diff represents the difference between sentences. We will mention the details of the function diff and the *context* in the description of introduced features' calculation, because they are essentially identical.

Because the context is composed of candidate translation of other sentences within a document, any candidate translation will be evaluated according to two aspects: the similarity between its own reference and itself (S_{eval}), and the relation with the other references where it becomes a context (D_{eval}). The former evaluation is preformed in a strict n-gram matching method to control the local translation of every word, whereas the latter is quite sketchy to reveal more sentence cross-relation in the document.

3.4 Document-Level Features

As a baseline, SMT system has already been tuned to optimize the S_{eval} (i.e., BLEU), we introduce two features, f_{doc}^t and f_{doc}^{st}, to represent the performance against D_{eval} as follows.

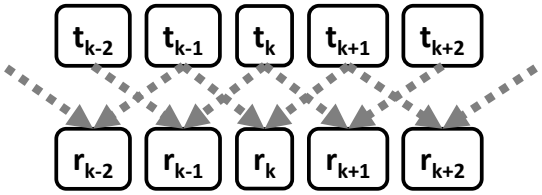

Figure 3: *In D_{eval}, every translation t will be compared with reference r of the other translations, where t becomes context within a document. Here, the context of t_k is t_{k-1} and t_{k+1}.*

$$f^t_{doc}(C) = \underset{1 \le k \le length(C)}{\text{avg}} \text{diff}(\text{context of } t_k \in C, t_k \in C) \tag{3}$$

Exp. (3) is similar to Exp. (2), with the $r_k \in D^r$ substituted for $t_k \in C$ (i.e., in Fig. 3, the lower rank and the upper rank are identical). The feature f^t_{doc} reveals the difference of a candidate translation with its context, which is also composed of candidate translations. Specifically, we take the context of t_k as:

$$\{t_{max(0,k-x)}, \cdots t_{k-1}, t_{k+1}, \cdots t_{min(length(C),k+x)}\} \tag{4}$$

Here x is a window size. Further, for the $\text{diff}(\cdot,\cdot)$ function, we want it to be flexible to reveal more sentence cross-relation as the strict lexical-based evaluation will be controlled by the S_{eval}. Therefore, we use the word-embedding technique to transform the lexical information into vector representation and use the distance between vectors as the diff function. Specifically, we define the $\text{diff}(\cdot,\cdot)$ as:

$$\text{diff}(\mu, \nu) = \log \|\mu - \nu\| \tag{5}$$

where μ and ν are two vectors and $\|\cdot\|$ is the Euclidean norm. To get the vector of a sentence or a set of sentences, we use the bag-of-word approach to get the average vector of all the word vectors contained by the sentence(s).

The f^t_{doc} feature concerns only to the target-side translation candidates. If candidate translations are similar to their context on average in a document, the feature will be small and if they are not, the feature will be large. On the other hand, we also need a feature to reveal the consistency in translation that can connect the source side and target side. So we use the feature f^{st}_{doc} to reveal the consistency in translation.

$$f^{st}_{doc}(D, C) = \underset{v}{\text{avg}} \left\{ \log \frac{\sum_w \underset{(v,w)\in(D,C)}{\text{count}}(v, w)}{|\{w|(v, w) \in (D, C)\}|} \right\} \tag{6}$$

Here, v is a word on source side and w is a word on target-side; (v, w) is a translated word pair and $\text{count}(\cdot)$ is a count function. Exp. (6) is essentially an average of token-type log-ratio over a source-side word v (Fig. 4): i.e., for v, we count the total times it has been translated by $\sum_w \text{count}(v, w)$ and count how many types of target-side words it has been translated to by $|\{w|(v, w) \in (D, C)\}|$. If source-side words are consistently translated to one or a few certain target-side words on average, the feature will grow large; if not, the feature will be small.

Besides the two document-level features we introduced, we also take the score generated by the baseline SMT system as a sentence-level feature f_{snt}. Then we use an interpolation of

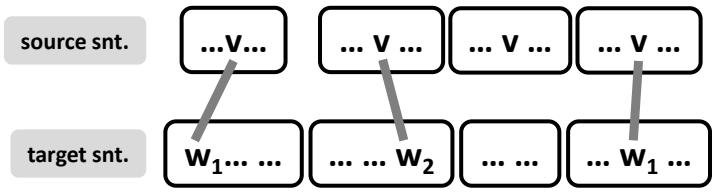

Figure 4: *Feature of f^{st}_{doc}. A source-side word v appears four times and is translated three times to two different target-side words w_1 and w_2. For v, we have a ratio of* $\log \frac{3}{2}$.

the features as the score for a $\{C_1, C_2, \cdots C_l\}$ search as follows.

$$\text{score}(\{D_1, \cdots, D_l\},\{C_1, \cdots, C_l\}) = \sum_{1 \le i \le l} \{\lambda_{snt} f_{snt}(D_i, C_i) + \lambda^t_{doc} f^t_{doc}(D_i, C_i) + \lambda^{st}_{doc} f^{st}_{doc}(D_i, C_i)\} \quad (7)$$
$$(|\lambda_{snt}| + |\lambda^t_{doc}| + |\lambda^{st}_{doc}| = 1)$$

3.5 Decoding and Tuning

The algorithm we used in decoding is basically a cube-pruning algorithm (Chiang, 2007) to merge the m-best list of translation candidates together over an entire document. Within the process of merging, the f^t_{doc} and f^{st}_{doc} are calculated. The merging needs to be conducted on the entire document because the f^{st}_{doc} can only be calculated for a given combination of sentence candidates over the entire document.[1]

Because the number of lists is equal to the number of sentences in a document, which usually becomes several tens or over a hundred, the original cube-pruning approach will not work well because its steps only forward to the next one candidate in each list from the present frontier, which prevents the search from generating enough combinations when there are too many lists. For example, consider a case in which we search 100 different combinations of sentence candidates over a document composed of 100 sentences; on average, we only touch the 2-best candidate (the one immediately below the top one) of each sentence. To avoid the problem, we use a wider margin B for each list in search rather than only $+1$ in the original algorithm.[2] The time complexity of the search for a document will be $O(N^2BT)$, where N is the number of sentences in a document; B is the width of the margin; and the T is the search times. For a document with N sentences, in each search, $N \cdot B$ combinations will be generated for f^t_{doc} and f^{st}_{doc} calculation. The two feature calculations are linear to the number of sentences in a document, i.e. $O(N)$. Thus, we have the described time complexity.

We apply the decoding algorithm on a development set of document pairs to tune the weights λ_{snt}, λ^t_{doc}, and λ^{st}_{doc}. According to Duh et al. (2012), the tuning algorithm is a multi-objective learning algorithm under the Pareto optimality. The method of Duh et al. (2012) is originally used for simultaneously tuning parameter weights to optimize different sentence-level translation measures. It has been shown that multi-objective tuning shows more robustness

[1] Note that we can set a window size for f^t_{doc}

[2] Specifically, the change is regarding line 11 of Fig. 6 in Chiang (2007). This line is executed multiple times in our search, with a more large enumerating margin for each list.

than traditional single-objective tuning. In our approach, we tune the weights under the Pareto optimality of $\{S_{eval}, D_{eval}\}$ as follows:

$$\underset{\lambda_{snt},\lambda^t_{doc},\lambda^{st}_{doc}}{\operatorname{argmax}} \{S_{eval}, +D_{eval}\} \qquad (\lambda^t_{doc} > 0) \tag{8}$$

$$\underset{\lambda_{snt},\lambda^t_{doc},\lambda^{st}_{doc}}{\operatorname{argmax}} \{S_{eval}, -D_{eval}\} \qquad (\lambda^t_{doc} < 0) \tag{9}$$

We maximize the S_{eval} (BLEU) because it is a measure for which the higher is the better. However, we are not sure in the case of D_{eval}. As mentioned, the D_{eval} and feature f^t_{doc} essentially have the same interpretation, so we make the sign of D_{eval} dependent on the sign of λ^t_{doc}, to make the optimization meaningful. When $\lambda^t_{doc} > 0$, i.e., the distance between a sentence and its context is to be encouraged, we maximize the D_{eval}; if the opposite, we minimize the D_{eval} (i.e., maximize the $-D_{eval}$).[3]

The multi-objective tuning will generate a Pareto frontier of multiple sets of weights rather than a single deterministic weight setting. The difference between the linear combination and Pareto optimality in multi-objective tuning has been discussed and compared in Duh et al. (2012). Generally, the Pareto optimality strategy is *to optimize first agnostically and a posteriori let the designer choose among a set of weights*. This philosophy is also reasonable in our approach, which is a post-process applied in a baseline SMT system to introduce document-level information. In practice, if document-level information is no available, our approach degenerates to the baseline system (i.e. $\lambda^t_{doc} = \lambda^{st}_{doc} = 0$); otherwise, the approach produces several sets of $\{\lambda^t_{doc}, \lambda^{st}_{doc}\}$, which suggests that we should pay attention to the document-level features.

4 Experiment

4.1 Data and Settings

We tested the proposed approach on French-to-English translation because this translation task has been handled well by state-of-the-art SMT systems. We used two different schemes. One is on the WIT[3] corpus of TED talks[4] (Cettolo et al., 2012), which contains a small training set with document-level parallel development set and test set. The other scheme is a relatively more realistic setting: using the Europarl corpus (Koehn, 2005) for model training and an in-domain development set for the weight tuning in the baseline SMT system. Then we selected document pairs from the Common Crawl (CC) Corpus[5] of WMT2013 for document-level development and test set. The CC corpus has a lot of noise, with many document pairs only several sentences long – too short for our purposes. Thus, we selected relatively high-quality document pairs, with moderate lengths of 40–60 sentences to compose the baseline. The data used in the two schemes and the detailed information are listed in Tables 1 and 2, respectively.

In experiments, as previously described, a baseline SMT system was built from sentence-level parallel data (the *train* row in Tables 1 and 2) and tuned on sentence-level development set (the *dev. (snt.)* row). We used the phrase-based statistical machine translation (PB SMT) system of Moses[6] (Koehn et al., 2007) as the baseline SMT system. In model training, we used the *grow-diag-final-and* to symmetrize the output of GIZA++[7] (Och and Ney, 2003). The *max-phrase-length* was set to 7 and the reordering model was *msd-bidirectional-fe*. The language

[3] We always set λ_{snt} to be positive. λ^t_{doc} and λ^{st}_{doc} can be either positive or negative.

[4] https://wit3.fbk.eu/

[5] http://www.statmt.org/wmt13/translation-task.html

[6] http://www.statmt.org/moses/

[7] https://code.google.com/p/giza-pp/

Table 1: *Data used in experiment.*

	scheme-1	scheme-2
train	WIT^3	Europarl
dev. (snt.)	WIT^3	WMT dev2006
dev. (doc.)	WIT^3	CC
test	WIT^3	CC

Table 2: *Number of sentence and document pairs of corpora. The dev. (snt.) and dev. (doc) of scheme-1are an identical set.*

	scheme-1	scheme-2
train	0.14M snt.	1.99M snt.
dev. (snt.)	934 snt.	2,000 snt.
dev. (doc.)	8 doc. / 934 snt.	14 doc. / 600 snt.
test	11 doc. / 1,664 snt.	55 doc. / 2,500 snt.

model was an interpolated 5-gram model with modified Kneser-Ney discounting, trained by SRILM[8] (Stolcke, 2002), on each scheme's training data. In sentence-level decoding, the *ttable-limit* was 20; the *stack* size was 200; and the *distortion-limit* was 6, all of which followed the default settings of Moses' decoder. The feature weights of the baseline PB SMT system were tuned by MERT (Och, 2003) to optimize the sentence-level development set BLEU (Papineni et al., 2002). The settings in tuning and translating on sentence-level were identical.

For the document-level decoding of the proposed approach, we used the baseline system to generate a 1000-best translation candidate list for each sentence in a document. Each translation candidate was attached with the word alignment information in sentence-level decoding for the f^{st}_{doc} calculation. Duplicate candidates in a 1000-best list were merged to one candidate taking the highest score[9] of the baseline SMT system. For the f^{t}_{doc} calculation, we used a high-quality English word embedding used in the SENNA[10] toolkit (Collobert et al., 2011).[11] The word embedding is over a vocabulary of 130,000 words, with 50-dimension vectors.

In the document-level decoding algorithm, we set the *margin* in cube-pruning to $[-10, 10]$ to enlarge the search space. The search generated 100 document-level candidates for re-ranking. In the f^{t}_{doc} feature and the D_{eval} calculation, we set *window-size* to 2. That is, the context was defined as the two preceding and two succeeding sentences.

For weight tuning on the document level, the multi-objective tuning can be combined with any tuning algorithm, such as MERT (Och, 2003), MIRA (Chiang, 2012), or PRO (Hopkins and May, 2011). Our approach contains only two free weights, λ^{t}_{doc} and λ^{st}_{doc}; thus, we used a *greedy search* for them in $(-1.0, 1.0)$, with step of 0.1, to avoid any possible search errors in the tuning phase.

We took the *consistency verification* approach (Xiao et al., 2011) as the comparison approach in our experiments. Similar to our approach, this approach takes advantage of the m-best translation candidates and uses a further decoding step to polish the baseline sentence-level

[8] `http://www.speech.sri.com/projects/srilm/`

[9] As well as the word alignment of the highest-scoring candidate.

[10] `http://ml.nec-labs.com/senna/`

[11] We also tried other vectors of word embedding, such as using `word2vec` (`https://code.google.com/p/word2vec/`) or `nplm` (`http://nlg.isi.edu/software/nplm/`) to train vectors on a data dump of Wikipedia. However, different vectors did not affect the performance much so we just used the pre-trained vectors of SENNA.

translation (the re-decoding in Xiao et al. (2011)). Specifically, the approach first generates a list of ambiguous words on the source side. Then it collects possible translations of those ambiguous words from m-best translation candidates and selects *one* standard translation for each ambiguous word. Finally, the translation model (i.e. phrase table) was filtered to ensure that it contains only the standard translation for ambiguous words. With the filtered translation model, a re-decoding is conducted.

In our experiment, we followed the instructions of Xiao et al. (2011), using 5-best list, a scaling factor α of 0.01,[12] and the M1 method, which leads to a better performance. A problem is that experiments conducted in Xiao et al. (2011) were on corpora of news, and they used a term database to select the source-side ambiguous word. Because we do not have such a resource and our experimental schemes have more variations, we selected the source-side ambiguous word by *tf-idf* score and took the top-k tf-idf words. We varied the k in the experiment.[13]

4.2 Results

In Table 3, we show the test set BLEU of the baseline SMT system and the effect of the consistency verification method. For scheme-1, the baseline achieve a test set BLEU of over 30, despite the scanty training data. In Cettolo et al. (2012), the performance on the same dataset of English-to-French is reported, which also had a test set BLEU of over 30. Because French and English have relatively similar vocabulary and syntax, we consider the baseline of scheme-1 reasonable. For scheme-2, the baseline's test set BLEU is also near to 30, as we intend to build a high baseline.[14] When we test the consistency verification method, we observe that it works on scheme-1 but not on scheme-2, and the performance worsens as when the number of verified words increases. We attribute the phenomenon to the rigidness of the consistency verification method. As mentioned, the data used in Xiao et al. (2011) are bound to the news field. Although the topics vary among the documents, a substantial consistency in special-term translation is required in the news field, and Xiao et al. (2011) did use a database of terms. However, the textual data used in our experiment are more casual and variable, especially in scheme-2. Consequently, the consistency verification method does not perform well in scheme-2.

In Tables 4 and 5, we show the experimental results of the proposed approach in scheme-1 and scheme-2, respectively. Different sets of weights on the frontier of Pareto optimality are listed,[15] with their corresponding S_{eval} and D_{eval} on development set and S_{eval} on test set (i.e., test set BLEU). The first rows, $\lambda^{t}_{doc} = 0$ and $\lambda^{st}_{doc} = 0$ are the performance of the baseline SMT system for scheme-1 and scheme-2. We conduct a statistical significance test via the *bootstrap* method (Koehn, 2004) using bleu-kit[16]. For each row, $+$ and $-$ mean the result is better or worse than the baseline, respectively: a single mark means the difference is on the level of $p < 0.05$ and a double mark means on the $p < 0.01$ level. For the overall performance, in scheme-1, the change of test set BLEU is in the range of $[-0.01, +0.48]$ points compared to the baseline; in scheme-2, the range of change is in $[-0.26, +0.56]$. Because the Pareto frontier offers multiple weights rather than a deterministic, the change on test set BLEU we report here is a range rather than a deterministic value.

[12] Xiao et al. (2011) said a proper α is in $[0.005, 0.1)$.

[13] The k is used to generate a list of source-side words for verification. The list also contains unambiguous words; hence, the types of verified source-side words are less than k in our experiment.

[14] The test set used in Hasler et al. (2014), which is chosen from the same CC data, has a baseline test set BLEU near 20.

[15] In search, we filter out the weights that noticeably worsen the development set S_{eval} (BLEU), which are worse than -0.5 point than the baseline.

[16] `http://www.nlp.mibel.cs.tsukuba.ac.jp/bleu_kit/`

Table 3: *Test set BLEU of baseline SMT system and consistency verification (Xiao et al., 2011) of* $k = 10, 20, 50$.

	scheme-1	scheme-2
baseline	31.10	29.39
top-10	31.17	28.46
top-20	31.08	28.21
top-50	30.22	27.13

Table 4: S_{eval} *and* D_{eval} *on development set and* S_{eval} *on test set (test set BLEU) in scheme-1. Different* λ_{doc}^{t} *and* λ_{doc}^{st} *are generated by multi-objective tuning.*

λ_{doc}^{t}	λ_{doc}^{st}	dev. S_{eval}	dev. D_{eval}	test BLEU
0.0	0.0	28.09	.2223	31.10
0.0	-0.2	28.10	.2224	31.09
$+0.2$	-0.4	27.76	.2238	31.50^{++}
$+0.2$	-0.5	27.70	.2242	31.54^{++}
$+0.3$	-0.3	27.69	.2243	31.50^{++}
$+0.3$	-0.4	27.64	.2250	31.58^{++}
$+0.4$	0.0	27.84	.2238	31.34^{++}
$+0.4$	-0.1	27.70	.2242	31.41^{++}
$+0.5$	-0.1	27.60	.2255	31.51^{++}

Table 5: S_{eval} *and* D_{eval} *on development set and* S_{eval} *on test set (test set BLEU) in scheme-2. Different* λ_{doc}^{t} *and* λ_{doc}^{st} *are generated by multi-objective tuning.*

λ_{doc}^{t}	λ_{doc}^{st}	dev. S_{eval}	dev. D_{eval}	test BLEU
0.0	0.0	28.34	.1583	29.39
-0.1	-0.8	28.79	.1566	29.95^{++}
-0.2	-0.6	28.73	.1565	29.89^{++}
-0.3	-0.5	28.62	.1562	29.85^{++}
-0.3	-0.6	28.38	.1549	29.71^{++}
-0.4	-0.4	28.51	.1557	29.80^{++}
-0.5	-0.4	28.34	.1539	29.45
-0.6	-0.1	28.44	.1556	29.76^{++}
-0.6	-0.3	28.28	.1534	29.39
-0.7	0.0	28.43	.1553	29.67^{++}
-0.7	-0.2	28.19	.1534	29.24
-0.9	0.0	28.11	.1532	29.13^{-}

5 Discussion

From the experimental results, we observe that the proposed approach can generate better results by introducing document-level features on different datasets. In Table 4, we observe that the development set S_{eval} in scheme-1 is actually decreased by different weights while the test set BLEU increases. This is because we use an identical development set for sentence-level tuning in the baseline SMT system and for the purposes of document-level tuning. Apparently, the tuning in the baseline system tends to over-fit the development set, and the proposed approach

Table 6: *Change on the test set BLEU of sentence-level (only changed translations are counted).*

	unchanged	increased	decreased	total
scheme-1	13.58%	12.32%	9.13%	35.03%
scheme-2	9.48%	13.16%	7.24%	29.88%

Table 7: *Change on the test set BLEU of document-level.*

	unchanged	increased	decreased	total
scheme-1	0 doc.	8 doc.	3 doc.	11 doc.
scheme-2	0 doc.	49 doc.	6 doc.	55 doc.

can release it by the D_{eval}. In Table 5, for scheme-2, we can observe both the development set S_{eval} and test set BLEU increase with most of the weights. In this scheme, the training data (including sentence-level tuning data) are quite different from the document-level development and test set. Hence, the efficiency of the document-level features are more obvious.

Compared with the consistency verification method, our approach uses no precise lexical features, which we rely on the baseline system to address. As a result, the proposed approach can avoid the rigidness of consistency verification and be adaptable to variant datasets.

For a further investigation, judging by their best performance,[17] we calculate the test set BLEU for each sentence and for each document in the two schemes. The sentence-by-sentence evaluation is shown in Table 6. We find that approximately one third of the sentences have been changed from the baseline and only approximately one eighth of the sentences see an improved BLEU. Figs. 5 and 6 depict the difference of BLEU of changed sentences. The document-by-document evaluation is shown in Table 7 and depicted in Figs. 7 and 8. We can observe that most documents in each scheme have an improvement of the BLEU score. The phenomenon suggests that document-level information does disturb the performance of special sentences (as the "*risky*" stated in Tiedemann (2010)) because the baseline SMT system has already done a good job. However, treating the document as an evaluation unit can lead to better performance.

We show a translation example in Table 8. In the example, the French word *voyage* is selected to be verified and its translation is fixed to be *journey*. This is not a wrong translation, although more variations are usually required. On the other hand, the French word *ville* is translated to *town* in the baseline and untouched by the consistency verification method, whereas the proposed approach can select a more correct translation of *city*. We can see the proposed approach to be a more flexible approach than consistency verification.

6 Conclusions and Future Work

In this paper, we introduced two document-level features to improve a sentence-level baseline SMT system. In the proposed approach, we integrated document-level information to the sentence-level translation using multi-objective tuning under both a sentence- and document-level measure. Experimental results demonstrated that the approach works on different datasets and experimental schemes.

We plan to explore more document-level features and improve the search algorithm in future. We are considering applying the linear programming method in Koshikawa et al. (2010) to our document-level decoding.

[17] $\lambda^t_{doc} = +0.3$, $\lambda^{st}_{doc} = -0.4$ for scheme-1; $\lambda^t_{doc} = -0.1$, $\lambda^{st}_{doc} = -0.8$ for scheme-2.

Table 8: *Translation example of baseline, consistency verification, and our proposed approach.*

Input
le voyage doit débute et se terminer dans le même pays , mais pas forcément dans la même ville .
Reference
· travel must begin and end in the same country , but not necessarily in the same city .
Baseline
the trip must begin and end in the same country , but not necessary in the same town .
Consistency Verification
the journey must begin and end in the same country , but not necessary in the same town .
Proposed Approach
the trip must begin and end in the same country , but not necessary in the same city .

Figure 5: *Change on test set BLEU of sentence-level translation, best output of scheme-1, sorted by the difference (only changed translations are illustrated).*

Figure 6: *Change on test set BLEU of sentence-level translation, best output of scheme-2, sorted by the difference (only changed translations are illustrated).*

Figure 7: *Change on test set BLEU of document-level translation, best output of scheme-1, sorted by the difference.*

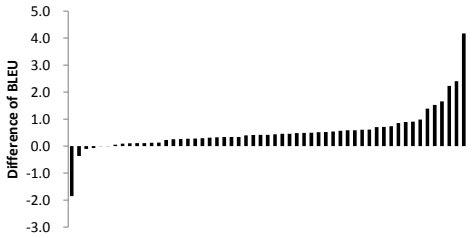

Figure 8: *Change on test set BLEU of document-level translation, best output of scheme-2, sorted by the difference.*

References

Ben, G., Xiong, D., Teng, Z., Lü, Y., and Liu, Q. (2013). Bilingual lexical cohesion trigger model for document-level machine translation. In *Proceedings of the 51st Annual Meeting of the Association for*

Computational Linguistics (Volume 2: Short Papers), pages 382–386, Sofia, Bulgaria.

Carpuat, M. (2009). One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 19–27, Boulder, Colorado.

Carpuat, M. and Simard, M. (2012). The trouble with SMT consistency. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 442–449, Montréal, Canada.

Cettolo, M., Girardi, C., and Federico, M. (2012). WIT^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Chiang, D. (2012). Hope and fear for discriminative training of statistical translation models. *The Journal of Machine Learning Research*, 13:1159–1187.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Duh, K., Sudoh, K., Wu, X., Tsukada, H., and Nagata, M. (2012). Learning to translate with multiple objectives. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Jeju Island, Korea.

Eidelman, V., Boyd-Graber, J., and Resnik, P. (2012). Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 115–119, Jeju Island, Korea.

Gong, Z., Zhang, M., and Zhou, G. (2011). Cache-based document-level statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 909–919, Edinburgh, Scotland, UK.

Guillou, L. (2013). Analysing lexical consistency in translation. In *Proceedings of the Workshop on Discourse in Machine Translation*, pages 10–18, Sofia, Bulgaria.

Hardmeier, C., Nivre, J., and Tiedemann, J. (2012). Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190, Jeju Island, Korea.

Hasler, E., Blunsom, P., Koehn, P., and Haddow, B. (2014). Dynamic topic adaptation for phrase-based mt. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–337, Gothenburg, Sweden.

Hopkins, M. and May, J. (2011). Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the 10th Machine Translation Summit*, pages 79–86, Phuket, Thailand.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Koshikawa, M., Utiyama, M., Umetani, S., Matsui, T., and Yamamoto, M. (2010). N-best reranking using optimal phrase alignment for statistical machine translation. *Journal of Information Processing*, 51:1443–1451. (in Japanese).

Meyer, T. and Webber, B. (2013). Implicitation of discourse connectives in (machine) translation. In *Proceedings of the Workshop on Discourse in Machine Translation*, pages 19–26, Sofia, Bulgaria.

Och, F. J. (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.

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

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (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.

Stolcke, A. (2002). SRILM–an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing*, pages 901–904, Denver, Colorado.

Stymne, S., Hardmeier, C., Tiedemann, J., and Nivre, J. (2013). Feature weight optimization for discourse-level SMT. In *Proceedings of the Workshop on Discourse in Machine Translation*, pages 60–69, Sofia, Bulgaria.

Tiedemann, J. (2010). Context adaptation in statistical machine translation using models with exponentially decaying cache. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 8–15, Uppsala, Sweden.

Ture, F., Oard, D. W., and Resnik, P. (2012). Encouraging consistent translation choices. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 417–426, Montréal, Canada.

Xiao, T., Zhu, J., Yao, S., and Zhang, H. (2011). Document-level consistency verification in machine translation. In *Proceedings of the 13th Machine Translation Summit*, pages 131–138, Xiamen, China.

Xiong, D., Ben, G., Zhang, M., Lü, Y., and Liu, Q. (2013a). Modeling lexical cohesion for document-level machine translation. In *Proceedings of 23rd International Conference on Artificial Intelligence*, pages 2183–2189, Beijing, China.

Xiong, D., Ding, Y., Zhang, M., and Tan, C. L. (2013b). Lexical chain based cohesion models for document-level statistical machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1563–1573, Seattle, Washington, USA.

Xiong, D. and Zhang, M. (2013). A topic-based coherence model for statistical machine translation. In *Proceedings of the the 27th AAAI Conference on Artificial Intelligence*, pages 977–983, Bellevue, Washington, USA.