Bilingually Motivated Domain-Adapted Word Segmentation for Statistical Machine Translation

Yanjun Ma **Andy Way**
National Centre for Language Technology
School of Computing
Dublin City University
Dublin 9, Ireland
{yma, away}@computing.dcu.ie

Abstract

We introduce a word segmentation approach to languages where word boundaries are not orthographically marked, with application to Phrase-Based Statistical Machine Translation (PB-SMT). Instead of using manually segmented *monolingual* domain-specific corpora to train segmenters, we make use of bilingual corpora and statistical word alignment techniques. First of all, our approach is adapted for the specific translation task at hand by taking the corresponding source (target) language into account. Secondly, this approach does not rely on manually segmented training data so that it can be automatically adapted for different domains. We evaluate the performance of our segmentation approach on PB-SMT tasks from two domains and demonstrate that our approach scores consistently among the best results across different data conditions.

1 Introduction

State-of-the-art Statistical Machine Translation (SMT) requires a certain amount of bilingual corpora as training data in order to achieve competitive results. The only assumption of most current statistical models (Brown et al., 1993; Vogel et al., 1996; Deng and Byrne, 2005) is that the aligned sentences in such corpora should be segmented into sequences of tokens that are meant to be words. Therefore, for languages where word boundaries are not orthographically marked, tools which segment a sentence into words are required. However, this segmentation is normally performed as a preprocessing step using various word segmenters. Moreover, most of these segmenters are usually trained on a manually segmented domain-specific corpus, which is not adapted for the specific translation task at hand given that the manual segmentation is performed in a *monolingual* context. Consequently, such segmenters cannot produce consistently good results when used across different domains.

A substantial amount of research has been carried out to address the problems of word segmentation. However, most research focuses on combining various segmenters either in SMT training or decoding (Dyer et al., 2008; Zhang et al., 2008). One important yet often neglected fact is that the optimal segmentation of the source (target) language is dependent on the target (source) language itself, its domain and its genre. Segmentation considered to be "good" from a *monolingual* point of view may be unadapted for training alignment models or PB-SMT decoding (Ma et al., 2007). The resulting segmentation will consequently influence the performance of an SMT system.

In this paper, we propose a bilingually motivated automatically domain-adapted approach for SMT. We utilise a small bilingual corpus with the relevant language segmented into basic writing units (e.g. characters for Chinese or kana for Japanese). Our approach consists of using the output from an existing statistical word aligner to obtain a set of candidate "words". We evaluate the reliability of these candidates using simple metrics based on co-occurrence frequencies, similar to those used in associative approaches to word alignment (Melamed, 2000). We then modify the segmentation of the respective sentences in the parallel corpus according to these candidate words; these modified sentences are then given back to the word aligner, which produces new alignments. We evaluate the validity of our approach by measuring the influence of the segmentation process on Chinese-to-English Machine Translation (MT) tasks in two different domains.

The remainder of this paper is organised as fol-

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 549–557,
Athens, Greece, 30 March – 3 April 2009.

lows. In section 2, we study the influence of word segmentation on PB-SMT across different domains. Section 3 describes the working mechanism of our bilingually motivated word segmentation approach. In section 4, we illustrate the adaptation of our decoder to this segmentation scheme. The experiments conducted in two different domains are reported in Section 5 and 6. We discuss related work in section 7. Section 8 concludes and gives avenues for future work.

2 The Influence of Word Segmentation on SMT: A Pilot Investigation

The *monolingual* word segmentation step in traditional SMT systems has a substantial impact on the performance of such systems. A considerable amount of recent research has focused on the influence of word segmentation on SMT (Ma et al., 2007; Chang et al., 2008; Zhang et al., 2008); however, most explorations focused on the impact of various segmentation guidelines and the mechanisms of the segmenters themselves. A current research interest concerns consistency of performance across different domains. From our experiments, we show that monolingual segmenters cannot produce consistently good results when applied to a new domain.

Our pilot investigation into the influence of word segmentation on SMT involves three off-the-shelf Chinese word segmenters including ICTCLAS (ICT) Olympic version[1], LDC segmenter[2] and Stanford segmenter version 2006-05-11[3]. Both ICTCLAS and Stanford segmenters utilise machine learning techniques, with Hidden Markov Models for ICT (Zhang et al., 2003) and conditional random fields for the Stanford segmenter (Tseng et al., 2005). Both segmentation models were trained on news domain data with named entity recognition functionality. The LDC segmenter is dictionary-based with word frequency information to help disambiguation, both of which are collected from data in the news domain. We used Chinese character-based and manual segmentations as contrastive segmentations. The experiments were carried out on a range of data sizes from news and dialogue domains using a state-of-the-art Phrase-Based SMT (PB-SMT) system—Moses (Koehn et al., 2007). The performance of PB-SMT system is measured with BLEU score (Papineni et al., 2002).

We firstly measure the influence of word segmentation on in-domain data with respect to the three above mentioned segmenters, namely UN data from the NIST 2006 evaluation campaign. As can be seen from Table 1, using monolingual segmenters achieves consistently better SMT performance than character-based segmentation (CS) on different data sizes, which means character-based segmentation is not good enough for this domain where the vocabulary tends to be large. We can also observe that the ICT and Stanford segmenter consistently outperform the LDC segmenter. Even using 3M sentence pairs for training, the differences between them are still statistically significant ($p < 0.05$) using approximate randomisation (Noreen, 1989) for significance testing.

	40K	**160K**	**640K**	**3M**
CS	8.33	12.47	14.40	17.80
ICT	10.17	14.85	**17.20**	20.50
LDC	9.37	13.88	15.86	19.59
Stanford	**10.45**	**15.26**	16.94	**20.64**

Table 1: Word segmentation on NIST data sets

However, when tested on out-of-domain data, i.e. IWSLT data in the dialogue domain, the results seem to be more difficult to predict. We trained the system on different sizes of data and evaluated the system on two test sets: IWSLT 2006 and 2007. From Table 2, we can see that on the IWSLT 2006 test sets, LDC achieves consistently good results and the Stanford segmenter is the worst.[4] Furthermore, the character-based segmentation also achieves competitive results. On IWSLT 2007, all monolingual segmenters outperform character-based segmentation and the LDC segmenter is only slightly better than the other segmenters.

From the experiments reported above, we can reach the following conclusions. First of all, character-based segmentation cannot achieve state-of-the-art results in most experimental SMT settings. This also motivates the necessity to work on better segmentation strategies. Second, monolingual segmenters cannot achieve consis-

[1] http://ictclas.org/index.html

[2] http://www.ldc.upenn.edu/Projects/Chinese

[3] http://nlp.stanford.edu/software/segmenter.shtml

[4] Interestingly, the developers themselves also note the sensitivity of the Stanford segmenter and incorporate external lexical information to address such problems (Chang et al., 2008).

		40K	160K
IWSLT06	CS	19.31	23.06
	Manual	19.94	-
	ICT	20.34	23.36
	LDC	**20.37**	**24.34**
	Stanford	18.25	21.40
IWSLT07	CS	29.59	30.25
	Manual	**33.85**	-
	ICT	31.18	33.38
	LDC	31.74	**33.44**
	Stanford	30.97	33.41

Table 2: Word segmentation on IWSLT data sets

tently good results when used in another domain. In the following sections, we propose a bilingually motivated segmentation approach which can be automatically derived from a small representative data set and the experiments show that we can consistently obtain state-of-the-art results in different domains.

3 Bilingually Motivated Word Segmentation

3.1 Notation

While in this paper, we focus on Chinese–English, the method proposed is applicable to other language pairs. The notation, however, assumes Chinese–English MT. Given a Chinese sentence c_1^J consisting of J characters $\{c_1, \ldots, c_J\}$ and an English sentence e_1^I consisting of I words $\{e_1, \ldots, e_I\}$, $A_{C \to E}$ will denote a Chinese-to-English word alignment between c_1^J and e_1^I. Since we are primarily interested in 1-to-n alignments, $A_{C \to E}$ can be represented as a set of pairs $a_i = \langle C_i, e_i \rangle$ denoting a link between one single English word e_i and a few Chinese characters C_i.The set C_i is empty if the word e_i is not aligned to any character in c_1^J.

3.2 Candidate Extraction

In the following, we assume the availability of an automatic word aligner that can output alignments $A_{C \to E}$ for any sentence pair (c_1^J, e_1^I) in a parallel corpus. We also assume that $A_{C \to E}$ contain 1-to-n alignments. Our method for Chinese word segmentation is as follows: whenever a single English word is aligned with several consecutive Chinese characters, they are considered candidates for grouping. Formally, given an alignment $A_{C \to E}$ between c_1^J and e_1^I, if $a_i = \langle C_i, e_i \rangle \in A_{C \to E}$, with $C_i = \{c_{i_1}, \ldots, c_{i_m}\}$ and $\forall k \in [\![1, m-1]\!]$, $i_{k+1} - i_k = 1$, then the alignment a_i between e_i and the sequence of words C_i is considered a candidate word. Some examples of such 1-to-n alignments between Chinese and English we can derive automatically are displayed in Figure 1.[5]

may	可 能	favorite	最 喜 欢
may	可 以	interesting	有 意 思
food	食 物	miami	迈 阿 密
food	食 品	last	最 后 一
july	七 月	block	个 街 区

Figure 1: Example of 1-to-n word alignments between English words and Chinese characters

3.3 Candidate Reliability Estimation

Of course, the process described above is error-prone, especially on a small amount of training data. If we want to change the input segmentation to give to the word aligner, we need to make sure that we are not making harmful modifications. We thus additionally evaluate the reliability of the candidates we extract and filter them before inclusion in our bilingual dictionary. To perform this filtering, we use two simple statistical measures. In the following, $a_i = \langle C_i, e_i \rangle$ denotes a candidate.

The first measure we consider is co-occurrence frequency ($COOC(C_i, e_i)$), i.e. the number of times C_i and e_i co-occur in the bilingual corpus. This very simple measure is frequently used in associative approaches (Melamed, 2000). The second measure is the alignment confidence (Ma et al., 2007), defined as

$$AC(a_i) = \frac{C(a_i)}{COOC(C_i, e_i)},$$

where $C(a_i)$ denotes the number of alignments proposed by the word aligner that are identical to a_i. In other words, $AC(a_i)$ measures how often the aligner aligns C_i and e_i when they co-occur. We also impose that $|C_i| \leq k$, where k is a fixed integer that may depend on the language pair (between 3 and 5 in practice). The rationale behind this is that it is very rare to get reliable alignments between one word and k consecutive words when k is high.

[5]While in this paper we are primarily concerned with languages where the word boundaries are not orthographically marked, this approach, however, can also be applied to languages marked with word boundaries to construct *bilingually* motivated "words".

The candidates are included in our bilingual dictionary if and only if their measures are above some fixed thresholds t_{COOC} and t_{AC}, which allow for the control of the size of the dictionary and the quality of its contents. Some other measures (including the Dice coefficient) could be considered; however, it has to be noted that we are more interested here in the filtering than in the discovery of alignments *per se*, since our method builds upon an existing aligner. Moreover, we will see that even these simple measures can lead to an improvement in the alignment process in an MT context.

3.4 Bootstrapped word segmentation

Once the candidates are extracted, we perform word segmentation using the bilingual dictionaries constructed using the method described above; this provides us with an updated training corpus, in which some character sequences have been replaced by a single token. This update is totally naive: if an entry $a_i = \langle C_i, e_i \rangle$ is present in the dictionary and matches one sentence pair (c_1^J, e_1^I) (i.e. C_i and e_i are respectively contained in c_1^J and e_1^I), then we replace the sequence of characters C_i with a single token which becomes a new lexical unit.[6] Note that this replacement occurs even if no alignment was found between C_i and e_i for the pair (c_1^J, e_1^I). This is motivated by the fact that the filtering described above is quite conservative; we trust the entry a_i to be correct.

This process can be applied several times: once we have grouped some characters together, they become the new basic unit to consider, and we can re-run the same method to get additional groupings. However, we have not seen in practice much benefit from running it more than twice (few new candidates are extracted after two iterations).

4 Word Lattice Decoding

4.1 Word Lattices

In the decoding stage, the various segmentation alternatives can be encoded into a compact representation of word lattices. A word lattice $G = \langle V, E \rangle$ is a directed acyclic graph that formally is a weighted finite state automaton. In the case of word segmentation, each edge is a candidate word associated with its weights. A straightforward estimation of the weights is to distribute the probability mass for each node uniformly to each outgoing edge. The single node having no outgoing edges is designated the "end node". An example of word lattices for a Chinese sentence is shown in Figure 2.

[6] In case of overlap between several groups of words to replace, we select the one with the highest confidence (according to t_{AC}).

4.2 Word Lattice Generation

Previous research on generating word lattices relies on multiple *monolingual* segmenters (Xu et al., 2005; Dyer et al., 2008). One advantage of our approach is that the bilingually motivated segmentation process facilitates word lattice generation without relying on other segmenters. As described in section 3.4, the update of the training corpus based on the constructed *bilingual* dictionary requires that the sentence pair meets the bilingual constraints. Such a segmentation process in the training stage facilitates the utilisation of word lattice decoding.

4.3 Phrase-Based Word Lattice Decoding

Given a Chinese input sentence c_1^J consisting of J characters, the traditional approach is to determine the best word segmentation and perform decoding afterwards. In such a case, we first seek a single best segmentation:

$$\hat{f}_1^K = \arg\max_{f_1^K, K} \{Pr(f_1^K | c_1^J)\}$$

Then in the decoding stage, we seek:

$$\hat{e}_1^I = \arg\max_{e_1^I, I} \{Pr(e_1^I | \hat{f}_1^K)\}$$

In such a scenario, some segmentations which are potentially optimal for the translation may be lost. This motivates the need for word lattice decoding. The search process can be rewritten as:

$$\begin{aligned}
\hat{e}_1^I &= \arg\max_{e_1^I, I} \{\max_{f_1^K, K} Pr(e_1^I, f_1^K | c_1^J)\} \\
&= \arg\max_{e_1^I, I} \{\max_{f_1^K, K} Pr(e_1^I) Pr(f_1^K | e_1^I, c_1^J)\} \\
&= \arg\max_{e_1^I, I} \{\max_{f_1^K, K} Pr(e_1^I) Pr(f_1^K | e_1^I) Pr(f_1^K | c_1^J)\}
\end{aligned}$$

Given the fact that the number of segmentations f_1^K grows exponentially with respect to the number of characters K, it is impractical to firstly enumerate all possible f_1^K and then to decode. However, it is possible to enumerate all the alternative segmentations for a substring of c_1^J, making the utilisation of word lattices tractable in PB-SMT.

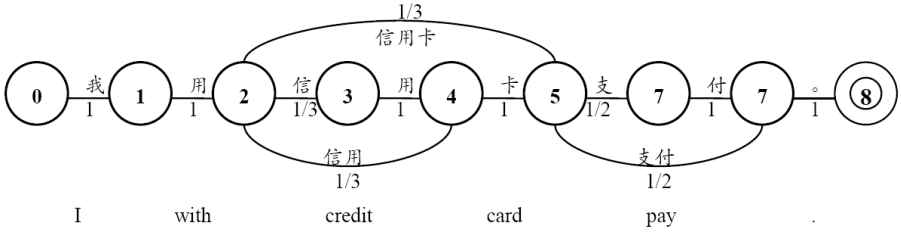

Figure 2: Example of a word lattice

5 Experimental Setting

5.1 Evaluation

The intrinsic quality of word segmentation is normally evaluated against a manually segmented gold-standard corpus using F-score. While this approach can give a direct evaluation of the quality of the word segmentation, it is faced with several limitations. First of all, it is really difficult to build a reliable and objective gold-standard given the fact that there is only 70% agreement between native speakers on this task (Sproat et al., 1996). Second, an increase in F-score does not necessarily imply an improvement in translation quality. It has been shown that F-score has a very weak correlation with SMT translation quality in terms of BLEU score (Zhang et al., 2008). Consequently, we chose to extrinsically evaluate the performance of our approach via the Chinese–English translation task, i.e. we measure the influence of the segmentation process on the final translation output. The quality of the translation output is mainly evaluated using BLEU, with NIST (Doddington, 2002) and METEOR (Banerjee and Lavie, 2005) as complementary metrics.

5.2 Data

The data we used in our experiments are from two different domains, namely news and travel dialogues. For the news domain, we trained our system using a portion of UN data for NIST 2006 evaluation campaign. The system was developed on LDC Multiple-Translation Chinese (MTC) Corpus and tested on MTC part 2, which was also used as a test set for NIST 2002 evaluation campaign.

For the dialogue data, we used the Chinese–English datasets provided within the IWSLT 2007 evaluation campaign. Specifically, we used the standard training data, to which we added devset1 and devset2. Devset4 was used to tune the parameters and the performance of the system was tested on both IWSLT 2006 and 2007 test sets. We used both test sets because they are quite different in terms of sentence length and vocabulary size. To test the scalability of our approach, we used HIT corpus provided within IWSLT 2008 evaluation campaign. The various statistics for the corpora are shown in Table 3.

5.3 Baseline System

We conducted experiments using different segmenters with a standard log-linear PB-SMT model: GIZA++ implementation of IBM word alignment model 4 (Och and Ney, 2003), the refinement and phrase-extraction heuristics described in (Koehn et al., 2003), minimum-error-rate training (Och, 2003), a 5-gram language model with Kneser-Ney smoothing trained with SRILM (Stolcke, 2002) on the English side of the training data, and Moses (Koehn et al., 2007; Dyer et al., 2008) to translate both single best segmentation and word lattices.

6 Experiments

6.1 Results

The initial word alignments are obtained using the baseline configuration described above by segmenting the Chinese sentences into characters. From these we build a bilingual 1-to-n dictionary, and the training corpus is updated by grouping the characters in the dictionaries into a single word, using the method presented in section 3.4. As previously mentioned, this process can be repeated several times. We then extract aligned phrases using the same procedure as for the baseline system; the only difference is the basic unit we are considering. Once the phrases are extracted, we perform the estimation of weights for the features of the log-linear model. We then use a simple dictionary-based maximum matching algorithm to obtain a single-best segmentation for the Chinese sentences in the development set so that

		Train		Dev.		Eval.	
		Zh	En	Zh	En	Zh	En
Dialogue	Sentences	40,958		489 (7 ref.)		489 (6 ref.)/489 (7 ref.)	
	Running words	488,303	385,065	8,141	46,904	8,793/4,377	51,500/23,181
	Vocabulary size	2,742	9,718	835	1,786	936/772	2,016/1,339
News	Sentences	40,000		993 (9 ref.)		878 (4 ref.)	
	Running words	1,412,395	956,023	41,466	267,222	38,700	105,530
	Vocabulary size	6057	20,068	1,983	10,665	1,907	7,388

Table 3: Corpus statistics for Chinese (Zh) character segmentation and English (En)

minimum-error-rate training can be performed.[7] Finally, in the decoding stage, we use the same segmentation algorithm to obtain the single-best segmentation on the test set, and word lattices can also be generated using the bilingual dictionary. The various parameters of the method (k, t_{COOC}, t_{AC}, cf. section 3) were optimised on the development set. One iteration of character grouping on the NIST task was found to be enough; the optimal set of values was found to be $k = 3$, $t_{AC} = 0.0$ and $t_{COOC} = 0$, meaning that all the entries in the bilingually dictionary are kept. On IWSLT data, we found that two iterations of character grouping were needed: the optimal set of values was found to be $k = 3$, $t_{AC} = 0.3$, $t_{COOC} = 8$ for the first iteration, and $t_{AC} = 0.2$, $t_{COOC} = 15$ for the second.

As can be seen from Table 4, our bilingually motivated segmenter (BS) achieved statistically significantly better results than character-based segmentation when enhanced with word lattice decoding.[8] Compared to the best in-domain segmenter, namely the Stanford segmenter on this particular task, our approach is inferior according to BLEU and NIST. We firstly attribute this to the small amount of training data, from which a high quality bilingual dictionary cannot be obtained due to data sparseness problems. We also attribute this to the vast amount of named entity terms in the test sets, which is extremely difficult for our approach.[9] We expect to see better results when a larger amount of data is used and the segmenter is enhanced with a named entity recogniser. On IWSLT data (cf. Tables 5 and 6), our approach yielded a consistently good performance on both translation tasks compared to the best in-domain segmenter—the LDC segmenter. Moreover, the good performance is confirmed by all three evaluation measures.

	BLEU	NIST	METEOR
CS	8.43	4.6272	0.3778
Stanford	**10.45**	**5.0675**	0.3699
BS-SingleBest	7.98	4.4374	0.3510
BS-WordLattice	9.04	4.6667	**0.3834**

Table 4: BS on NIST task

	BLEU	NIST	METEOR
CS	0.1931	6.1816	0.4998
LDC	0.2037	6.2089	0.4984
BS-SingleBest	0.1865	5.7816	0.4602
BS-WordLattice	**0.2041**	**6.2874**	**0.5124**

Table 5: BS on IWSLT 2006 task

	BLEU	NIST	METEOR
CS	0.2959	6.1216	0.5216
LDC	**0.3174**	6.2464	0.5403
BS-SingleBest	0.3023	6.0476	0.5125
BS-WordLattice	0.3171	**6.3518**	**0.5603**

Table 6: BS on IWSLT 2007 task

6.2 Parameter Search Graph

The reliability estimation process is computationally intensive. However, this can be easily parallelised. From our experiments, we observed that the translation results are very sensitive to the parameters and this search process is essential to achieve good results. Figure 3 is the search graph on the IWSLT data set in the first iteration step. From this graph, we can see that filtering of the bilingual dictionary is essential in order to achieve better performance.

[7] In order to save computational time, we used the same set of parameters obtained above to decode both the single-best segmentation and the word lattice.

[8] Note the BLEU scores are particularly low due to the number of references used (4 references), in addition to the small amount of training data available.

[9] As we previously point out, both ICT and Stanford segmenters are equipped with named entity recognition functionality. This may risk causing data sparseness problems on small training data. However, this is beneficial in the translation process compared to character-based segmentation.

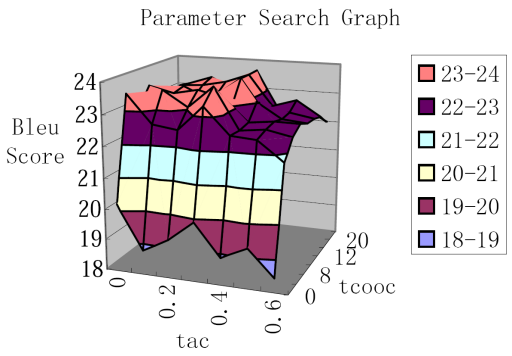

Figure 3: The search graph on development set of IWSLT task

6.3 Vocabulary Size

Our bilingually motivated segmentation approach has to overcome another challenge in order to produce competitive results, i.e. data sparseness. Given that our segmentation is based on bilingual dictionaries, the segmentation process can significantly increase the size of the vocabulary, which could potentially lead to a data sparseness problem when the size of the training data is small. Tables 7 and 8 list the statistics of the Chinese side of the training data, including the total vocabulary (Voc), number of character vocabulary (Char.voc) in Voc, and the running words (Run.words) when different word segmentations were used. From Table 7, we can see that our approach suffered from data sparseness on the NIST task, i.e. a large vocabulary was generated, of which a considerable amount of characters still remain as separate words. On the IWSLT task, since the dictionary generation process is more conservative, we maintained a reasonable vocabulary size, which contributed to the final good performance.

	Voc.	Char.voc	Run. Words
CS	6,057	6,057	1,412,395
ICT	16,775	1,703	870,181
LDC	16,100	2,106	881,861
Stanford	22,433	1,701	880,301
BS	**18,111**	**2,803**	**927,182**

Table 7: Vocabulary size of NIST task (40K)

6.4 Scalability

The experimental results reported above are based on a small training corpus containing roughly 40,000 sentence pairs. We are particularly interested in the performance of our segmentation approach when it is scaled up to larger amounts of data. Given that the optimisation of the bilingual dictionary is computationally intensive, it is impractical to directly extract candidate words and estimate their reliability. As an alternative, we can use the obtained bilingual dictionary optimised on the small corpus to perform segmentation on the larger corpus. We expect competitive results when the small corpus is a representative sample of the larger corpus and large enough to produce reliable bilingual dictionaries without suffering severely from data sparseness.

	Voc.	Char.voc	Run. Words
CS	2,742	2,742	488,303
ICT	11,441	1,629	358,504
LDC	9,293	1,963	364,253
Stanford	18,676	981	348,251
BS	**3,828**	**2,740**	**402,845**

Table 8: Vocabulary size of IWSLT task (40K)

As we can see from Table 9, our segmentation approach achieved consistent results on both IWSLT 2006 and 2007 test sets. On the NIST task (cf. Table 10), our approach outperforms the basic character-based segmentation; however, it is still inferior compared to the other in-domain monolingual segmenters due to the low quality of the bilingual dictionary induced (cf. section 6.1).

	IWSLT06	IWSLT07
CS	23.06	30.25
ICT	23.36	33.38
LDC	**24.34**	**33.44**
Stanford	21.40	33.41
BS-SingleBest	22.45	30.76
BS-WordLattice	24.18	32.99

Table 9: Scale-up to 160K on IWSLT data sets

	160K	640K
CS	12.47	14.40
ICT	14.85	**17.20**
LDC	13.88	15.86
Stanford	**15.26**	16.94
BS-SingleBest	12.58	14.11
BS-WordLattice	13.74	15.33

Table 10: Scalability of BS on NIST task

6.5 Using different word aligners

The above experiments rely on GIZA++ to perform word alignment. We next show that our approach is not dependent on the word aligner given that we have a conservative reliability estimation procedure. Table 11 shows the results obtained on the IWSLT data set using the MTTK alignment tool (Deng and Byrne, 2005; Deng and Byrne, 2006).

	IWSLT06	**IWSLT07**
CS	21.04	31.41
ICT	20.48	31.11
LDC	20.79	30.51
Stanford	17.84	29.35
BS-SingleBest	19.22	29.75
BS-WordLattice	**21.76**	**31.75**

Table 11: BS on IWSLT data sets using MTTK

7 Related Work

(Xu et al., 2004) were the first to question the use of word segmentation in SMT and showed that the segmentation proposed by word alignments can be used in SMT to achieve competitive results compared to using monolingual segmenters. Our approach differs from theirs in two aspects. Firstly, (Xu et al., 2004) use word aligners to reconstruct a (monolingual) Chinese dictionary and reuse this dictionary to segment Chinese sentences as other monolingual segmenters. Our approach features the use of a bilingual dictionary and conducts a different segmentation. In addition, we add a process which optimises the bilingual dictionary according to translation quality. (Ma et al., 2007) proposed an approach to improve word alignment by optimising the segmentation of both source and target languages. However, the reported experiments still rely on some monolingual segmenters and the issue of scalability is not addressed. Our research focuses on avoiding the use of monolingual segmenters in order to improve the robustness of segmenters across different domains.

(Xu et al., 2005) were the first to propose the use of word lattice decoding in PB-SMT, in order to address the problems of segmentation. (Dyer et al., 2008) extended this approach to hierarchical SMT systems and other language pairs. However, both of these methods require some monolingual segmentation in order to generate word lattices. Our approach facilitates word lattice generation given that our segmentation is driven by the bilingual dictionary.

8 Conclusions and Future Work

In this paper, we introduced a bilingually motivated word segmentation approach for SMT. The assumption behind this motivation is that the language to be segmented can be tokenised into basic writing units. Firstly, we extract 1-to-n word alignments using statistical word aligners to construct a bilingual dictionary in which each entry indicates a correspondence between one English word and n Chinese characters. This dictionary is then filtered using a few simple association measures and the final bilingual dictionary is deployed for word segmentation. To overcome the segmentation problem in the decoding stage, we deployed word lattice decoding.

We evaluated our approach on translation tasks from two different domains and demonstrate that our approach is (i) not as sensitive as monolingual segmenters, and (ii) that the SMT system using our word segmentation can achieve state-of-the-art performance. Moreover, our approach can easily be scaled up to larger data sets and achieves competitive results if the small data used is a representative sample.

As for future work, firstly we plan to integrate some named entity recognisers into our approach. We also plan to try our approach in more domains and on other language pairs (e.g. Japanese–English). Finally, we intend to explore the correlation between vocabulary size and the amount of training data needed in order to achieve good results using our approach.

Acknowledgments

This work is supported by Science Foundation Ireland (O5/IN/1732) and the Irish Centre for High-End Computing.[10] We would like to thank the reviewers for their insightful comments.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, MI.

[10] http://www.ichec.ie/

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232, Columbus, OH.

Yonggang Deng and William Byrne. 2005. HMM word and phrase alignment for statistical machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 169–176, Vancouver, BC, Canada.

Yonggang Deng and William Byrne. 2006. MTTK: An alignment toolkit for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 265–268, New York City, NY.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145, San Francisco, CA.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1012–1020, Columbus, OH.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 48–54, Edmonton, AL, Canada.

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 *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.

Yanjun Ma, Nicolas Stroppa, and Andy Way. 2007. Bootstrapping word alignment via word packing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 304–311, Prague, Czech Republic.

I. Dan Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience, New York, NY.

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

Franz 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.

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

Richard W Sproat, Chilin Shih, William Gale, and Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3):377–404.

Andrea Stolcke. 2002. SRILM – An extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, Denver, CO.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of Fourth SIGHAN Workshop on Chinese Language Processing*, pages 168–171, Jeju Island, Republic of Korea.

Stefan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 836–841, Copenhagen, Denmark.

Jia Xu, Richard Zens, and Hermann Ney. 2004. Do we need Chinese word segmentation for statistical machine translation? In *ACL SIGHAN Workshop 2004*, pages 122–128, Barcelona, Spain.

Jia Xu, Evgeny Matusov, Richard Zens, and Hermann Ney. 2005. Integrated Chinese word segmentation in statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 141–147, Pittsburgh, PA.

Huaping Zhang, Hongkui Yu, Deyi Xiong, and Qun Liu. 2003. HHMM-based Chinese lexical analyzer ICTCLAS. In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing*, pages 184–187, Sappora, Japan.

Ruiqiang Zhang, Keiji Yasuda, and Eiichiro Sumita. 2008. Improved statistical machine translation by multiple Chinese word segmentation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 216–223, Columbus, OH.