Bilingually Motivated Word Segmentation for SMT

YANJUN MA and ANDY WAY
National Centre for Language Technology
School of Computing
Dublin City University

November 29, 2010

Abstract

We introduce a bilingually motivated word segmentation approach to languages where word boundaries are not orthographically marked, with application to Phrase-Based Statistical Machine Translation (PB-SMT). Our approach is motivated from the insight that PB-SMT systems can be improved by optimising the input representation to reduce the predictive power of translation models. We firstly present an approach to optimise the existing segmentation of both source and target languages for PB-SMT and demonstrate the effectiveness of this approach using a Chinese–English MT task, i.e. to measure the influence of the segmentation on the performance of PB-SMT systems. We report a 5.44% relative increase in BLEU score and a consistent increase according to other metrics. We then generalise this method for Chinese word segmentation without relying on any segmenters and show that using our segmentation PB-SMT can achieve more consistent state-of-the-art performance across two domains. There are two main advantages of our approach. First of all, it is adapted to 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.

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 behind 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. Even for

a language like English, where spaces can offer an easy approximation to the minimal content-bearing units, an optimal segmentation is still required when analysing multi-word units, especially non-compositional compounds such as "kick the bucket" and "hot dog" Melamed [1997].

However, this segmentation is often performed in a *monolingual* context without any bilingual consideration, i.e. the segmentation of the source (target) language is performed regardless of the corresponding target (source) language at hand, which makes the word alignment task more difficult since different languages may realise the same concept using varying numbers of words (cf. Wu [1997]). This can generate a great deal of complexity for (*bilingual*) word alignment models if the corresponding texts are inappropriately segmented. Moreover, most segmenters are usually trained on a manually segmented domain-specific corpus. Therefore, such a segmentation tends to be sensitive to the domain of the data and 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 in the context of PB-SMT. Some statistical alignment models allow for 1-to-n word alignments for those reasons; however, they rarely question the monolingual tokenisation and the basic unit of the alignment process.[1] Morever, statistical alignment models assume a first-order dependency between alignment decisions in order to make the alignment process efficient. Some long-distance dependencies cannot be captured under such models. Some more recent research focuses on combining various segmenters either in SMT training Zhang et al. [2008] or decoding Dyer 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 a PB-SMT system, and a bilingually motivated segmentation is highly desirable for PB-SMT tasks.

This current work is an extension to our previous work on boostrapping word alignment via word packing Ma et al. [2007]. We conduct further experiments using a more established PB-SMT system and extend this approach to perform independent word segmentation. We focus on optimising the segmentation with the goals of (i) simplifying the task of automatic word aligners by *packing* several consecutive words together when we believe they correspond to a single word in the opposite language. By identifying enough such cases, we reduce the number of 1-to-n alignments, thus making the task of word alignment both easier and more natural; from an information-theoretic perspective, such a process reduces the predictive power of translation models Melamed [1997]; and (ii) capturing long-distance dependencies between alignment decisions in a incremental manner, i.e. we bootstrap the word packing and subsequently op-

[1] Interestingly, this is actually even the case for approaches that directly model alignments between phrases Marcu and Wong [2002], Birch et al. [2006].

timise the word segmentation based on its influence on SMT performance. We then generalise this method to produce a bilingually motivated automatically domain-adapted word segmentation approach for PB-SMT without relying on any existing word segmenters. We first utilise a small bilingual corpus with the relevant language segmented into basic writing units (e.g. characters for Chinese), and then cast the segmentation problem into an alignment problem. We also investigate various issues regarding scalability related to such a process.

Specifically, our approach consists of using the output from an existing statistical word aligner (e.g. GIZA++) to obtain a set of candidate 'words' to be packed. 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], Tiedemann [2003]. 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.

The remainder of this paper is organised as follows. In section 2, we study the interaction between word segmentation and word alignment. Firstly we investigate the influence of word segmentation on PB-SMT across different domains and demonstrate the necessity of refining word segmentation methods for PB-SMT in order to achieve consistently good results. We also discuss the particular word alignment issues related to Chinese–English which motivated our approach. Section 3 describes the working mechanism of our bilingually motivated word segmentation approach, namely word packing. In section 4, we describe the generalisation of our approach and its application to independent word segmentation. The corresponding experiments are reported in section 5 and 6. In section 7 we discuss related work in this direction. Section 8 concludes and gives avenues for future work.

2 Interaction Between Word Segmentation and Alignment

In this section, we first detail a pilot study on the influence of word segmentation on the performance of PB-SMT. Then we show that the pervasive 1-to-n alignments in Chinese–English word alignment motivate us to take advantage of the interaction between word segmentation and alignment in order to simplify the alignment task.

2.1 The Influence of Word Segmentation on PB-SMT

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 have 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,[2] LDC segmenter[3] and Stanford segmenter version 2006-05-11.[4] 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 points of contrast. Table 1 shows the pairwise F-measure of the automatic segmenters. On the IWSLT data set in the dialogue domain, we observed the strongest agreement between the LDC and ICT segmenters, which is even stronger than for Stanford and ICT segmenters. On NIST data, as expected, the Stanford and ICT segmenters agree more. On both data sets, the LDC and Stanford segmenters shows the greatest discrepancies.

		ICT	LDC	Stanford
IWSLT	ICT	100	94.45	93.80
	LDC	**94.45**	100	90.13
	Stanford	93.80	90.13	100
NIST	ICT	100	95.18	96.44
	LDC	95.18	100	93.38
	Stanford	**96.44**	93.38	100

Table 1: Pairwise F-measure between segmenters

We conducted MT experiments on a range of different-sized amounts of the above-mentioned data using a state-of-the-art PB-SMT system: Moses Koehn et al. [2007]. The performance of the PB-SMT system is measured via BLEU score Papineni et al. [2002].

We first 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.[5] As can be seen from Table 2, using monolin-

[2] `http://ictclas.org/index.html`

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

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

[5] Note that the UN data containing parliamentary documents is not exactly 'in-domain'; however, it's more similar to news domain compared to dialogues. We choose this corpora simply because of the availability of this corpora and its relatively large amount that enable us to test our approach in terms of scalability. We expect a better performance on 'strictly'

gual 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 segmenters consistently outperform the LDC segmenter. Even using 3M sentence pairs for training, the differences between the Stanford and LDC segmenters 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 2: 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 amounts of data and evaluated the system on two test sets: IWSLT 2006 and 2007. From Table 3, we can see that on the IWSLT 2006 test sets, LDC achieves consistently good results and the Stanford segmenter is the worst.[6] Furthermore, character-based segmentation also achieves competitive results. On IWSLT 2007 data, all monolingual segmenters outperform character-based segmentation and the LDC segmenter is only slightly better than the other segmenters.

		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 3: Word segmentation on IWSLT data sets

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 settings. This also motivates the necessity to

in-domain data using the ICT and Stanford segmenters.

[6]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].

work on better segmentation strategies. Second, monolingual segmenters cannot achieve consistently 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. Using this approach, we can either enhance the existing monolingual segmenter or directly perform word segmentation without relying on any monolingual segmenters.

2.2 The Case of 1-to-n Word Alignment

The same concept can be expressed in different languages using varying numbers of words; for example, a single Chinese word may frequently surface as a compound or a collocation in English given the large differences between the two languages. To quickly (and approximately) evaluate this phenomenon, we trained the statistical IBM word-alignment model 4 Brown et al. [1993][7] using the GIZA++ software Och and Ney [2003] for the following language pairs: Chinese–English (ZH–EN), Italian–English (IT–EN), and German–English (DE–EN), using the IWSLT 2007 corpus Takezawa et al. [2002], Paul [2006] for the first two language pairs, and the Europarl corpus Koehn [2005] for the last one. These asymmetric models produce alignments between one word and several words in both directions. Word segmentation was performed totally independently of the bilingual alignment process, i.e. it is done in a *monolingual* context. For European languages, we apply the maximum entropy-based tokeniser of OpenNLP;[8] the Chinese sentences were manually segmented Paul [2006].

In Table 4, we report the frequencies of the different types of alignments for the various languages and directions. As expected, the number of $1{:}\,n$ alignments with $n \neq 1$ is high for Chinese–English ($\simeq 40\%$), and significantly higher than for European languages. The case of 1-to-n alignments is, therefore, obviously an important issue when dealing with Chinese–English word alignment.[9] We can also observe that for all three language pairs, most of the n words involved in 1-to-n alignments are consecutive (con.).

To verify the above findings, we also list the statistics obtained from the manually aligned Chinese–English corpus as shown in Table 5. This manually aligned corpus is IWSLT devset 3, which contains 502 sentence pairs after cleaning Ma et al. [2008]. We observed similar distribution for $1{:}\,n$ ($n{>}1$) alignments. The main difference is that the automatic aligners tend to produce more $1{:}\,0$ alignments, while human annotators tend to generate more $m{:}\,n$ alignments.

[7] More specifically, we performed 5 iterations of Model 1, 5 iterations of HMM, 5 iterations of Model 3, and 5 iterations of Model 4.

[8] `http://opennlp.sourceforge.net/`.

[9] Note that a $1{:}\,0$ alignment may denote a failure to capture a $1{:}\,n$ alignment with $n > 1$.

	1:0	1:1	1:2		1:3		1:n ($n > 3$)	
			con.	non-con.	con.	non-con.	con.	non-con.
ZH–EN	22.19	59.60	9.92	1.69	3.06	1.24	1.02	1.28
EN–ZH	28.48	57.08	9.27	1.81	1.28	0.83	0.42	0.83
IT–EN	16.96	64.77	11.85	1.12	3.98	0.49	0.50	0.34
EN–IT	25.34	62.15	8.75	1.00	1.35	0.47	0.50	0.45
DE–EN	22.05	65.34	5.86	1.92	1.10	1.26	0.31	2.16
EN–DE	24.39	65.38	4.86	2.28	0.5	1.08	0.10	1.40

Table 4: Distribution of alignment types for different language pairs (%)

	1:0	1:1	1:2		1:3		1:n ($n > 3$)		m:n
			con.	non-con.	con.	non-con.	con.	non-con.	
ZH–EN	3.41	65.47	8.97	3.01	1.80	0.37	0.22	0.00	16.75
EN–ZH	2.66	66.90	10.43	2.13	0.70	0.06	0.10	0.06	17.12

Table 5: Distribution of alignment types for manually aligned Chinese–English corpus (%)

2.3 Notation

While in this paper we focus on Chinese–English, the proposed method is applicable to any language pair; even for closely related languages, we expect improvements to be seen. Notwithstanding the generality of the approach, in what follows we assume Chinese–English MT in order to explain our notation. Given a Chinese sentence c_1^J consisting of J words $\{c_1, \ldots, c_J\}$ and an English sentence e_1^I consisting of I words $\{e_1, \ldots, e_I\}$, $A_{C \to E}$ (resp. $A_{E \to C}$) will denote a Chinese-to-English (resp. an English-to-Chinese) 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 $a_j = \langle c_j, E_j \rangle$ denoting a link between one single Chinese word c_j and a few English words E_j (and similarly for $A_{E \to C}$). The set E_j is empty if the word c_j is not aligned to any word in e_1^I.

3 Bootstrapping Word Alignment via Word Packing

Our approach (cf. Ma et al. [2007]) consists of packing consecutive words together when we believe they correspond to a single word in the other language. This bilingually motivated packing of words changes the basic unit of the alignment process, and simplifies the task of automatic word alignment. We thus minimise the number of 1-to-n alignments in order to obtain more comparable segmentations in the two languages. In this section, we present an automatic method that builds upon the output from an existing automatic word aligner. More specifically, we (i) use a word aligner to obtain 1-to-n alignments, (ii) extract candidates for word packing, (iii) estimate the reliability of these candi-

dates, (iv) replace the groups of words to pack by a single token in the parallel corpus, and (v) re-iterate the alignment process using the updated corpus. The first three steps are performed in both directions, and produce two *bilingual dictionaries* (source–target and target–source) of groups of words to pack.

3.1 Candidate Extraction

In the following, we assume the availability of an automatic word aligner that can output alignments $A_{C\rightarrow E}$ and $A_{E\rightarrow C}$ for any sentence pair (c_1^J, e_1^I) in a parallel corpus. We also assume that $A_{C\rightarrow E}$ and $A_{E\rightarrow C}$ contain 1-to-n alignments. Our method for repacking words is very simple: whenever a single word is aligned with several consecutive words, they are considered candidates for repacking. Formally, given an alignment $A_{C\rightarrow E}$ between c_1^J and e_1^I, if $a_j = \langle c_j, E_j \rangle \in A_{C\rightarrow E}$, with $E_j = \{e_{j_1}, \ldots, e_{j_m}\}$ and $\forall k \in [1, m-1](m \geq 2)$, $j_{k+1} - j_k = 1$, then the alignment a_j between c_j and the sequence of words E_j is considered a candidate for word repacking. The same goes for $A_{E\rightarrow C}$. Some examples of such 1-to-n alignments between Chinese and English (in both directions) that we can derive automatically are displayed in Figure 1.

白葡萄酒: white wine	closest: 最 近
百货公司: department store	fifteen: 十 五
抱歉: excuse me	fine: 很 好
报警: call the police	flight: 次 航班
杯: cup of	get: 拿 到
必须: have to	here: 在 这里

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

3.2 Candidate Reliability Estimation

Of course, the process described above is error-prone and if we want to change the input to the word aligner, we need to make sure that we are not making harmful modifications.[10] 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, $a_j = \langle c_j, E_j \rangle$ denotes a candidate.

The first measure we consider is co-occurrence frequency ($COOC(c_j, E_j)$), i.e. the number of times c_j and E_j co-occur in the bilingual corpus. This very simple measure is frequently used in associative approaches Melamed [1997],

[10] Consequently, if we compare our approach to the problem of collocation identification, we may say that we are more interested in precision than recall Smadja et al. [1996]. However, note that our goal is not recognising specific sequences of words such as compounds or collocations; rather it is making (bilingually motivated) changes that simplify the alignment process.

Tiedemann [2003]. The second measure is the alignment confidence, defined as

$$AC(a_j) = \frac{C(a_j)}{COOC(c_j, E_j)}, \tag{1}$$

where $C(a_j)$ denotes the number of alignments proposed by the word aligner that are identical to a_j. In other words, $AC(a_j)$ measures how often the aligner aligns c_j and E_j when they co-occur. We also impose that $|E_j| \leq k$, 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 obtain a reliable alignment between one word and k consecutive words when k is high.

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, since our method builds upon existing aligners. Moreover, we will see that even these simple measures can lead to an improvement in the alignment process in an MT context (cf. Section 6).

3.3 Bootstrapped Word Repacking

Once the candidates are extracted, we repack the words in the bilingual dictionaries constructed using the method described above; this provides us with an updated training corpus, in which some word sequences have been replaced by a single token. This update is totally naive; if an entry $a_j = \langle c_j, E_j \rangle$ is present in the dictionary and matches one sentence pair (c_1^J, e_1^I) (i.e. c_j and E_j are respectively contained in c_1^J and e_1^I), then we replace the sequence of words E_j with a single token which becomes a new lexical unit.[11] Note that this replacement occurs even if no alignment was found between c_j and E_j 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 update is performed in both directions. It is then possible to run the word aligner using the updated (simplified) parallel corpus, in order to obtain new alignments. By performing a deterministic word packing, we avoid the computation of the fertility parameters associated with fertility-based models.

Word packing can be applied several times; once we have grouped some words 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).

It is also important to note that this process is bilingually motivated and strongly depends on the language pair. For example, *white wine*, *excuse me*, *call*

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

the police, and *cup of* (cf. Figure 1) translate respectively as *vin blanc*, *excusez-moi*, *appellez la police*, and *tasse de* in French. Those groupings would not be found for a language pair such as French–English, which is consistent with the fact that they are less useful for French–English than for Chinese–English in an MT perspective.

3.4 Word Unpacking and Phrase-Based SMT Decoding

The bidirectional grouping approach can improve the quality of alignment and correspondingly improve the quality of phrase extraction and the estimation of related parameters. In the decoding stage, given that the input is not packed and the language model is also trained on unpacked word segmentations, we need to undertake 'word unpacking' before estimating the parameters. The unpacking in PB-SMT is performed following the phrase extraction process. Specifically, in a log-linear PB-SMT system, the phrase translation probabilities and lexical distortion models are re-estimated based on relative frequencies; the lexical weighting probabilities are calculated based on the lexical translation distribution with word packing.

The unpacking step is particularly necessary in the context of bilingual word packing, i.e. both source and target sentences are packed, given that the language models are trained on texts without word packing. If we constrain the word packing process by only packing the source language, the word unpacking step could be avoided and word-lattice decoding could be utilised instead. (cf. section 4)

4 Bilingually Motivated Word Segmentation

4.1 Word Segmentation as an Alignment Problem

The approach proposed in section 3 can be applied to word segmentation by only packing the source language. The only assumption is that the sentence to be segmented can be split into basic writing units (e.g. characters for Chinese and kana for Japanese). The notation in section 2.3 can be easily adapted for this task. 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 character-to-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.

4.2 Bootstrapping Word Segmentation

We use the same approach proposed in section 3.1 to extract candidate words. 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. Some examples of such 1-to-n alignments between Chinese *characters* and English *words* we can derive automatically are displayed in Figure 2.

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

We can use the same measures proposed in section 3.2 to estimate the reliability of the candidate words and use the boostrapping approach in section 3.3 to derive better word segmentation.

4.3 Word Lattice Decoding

Casting word segmentation as an alignment problem implies that word segmentation of a sentence depends not only on the current sentence to segment but also on the corresponding target language. In such a context, the word lattice representation is particularly suitable in the decoding stage which aims to search for the most possible target sentence.

4.3.1 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.[12] 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 3.

4.3.2 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 4.2, the update of the training corpus based on the constructed *bilingual* dictionary

[12] We can also use language models to assign probabilities to each edge as in Xu et al. [2005]. In this case, however, we have to rely on some segmented data to train the language model.

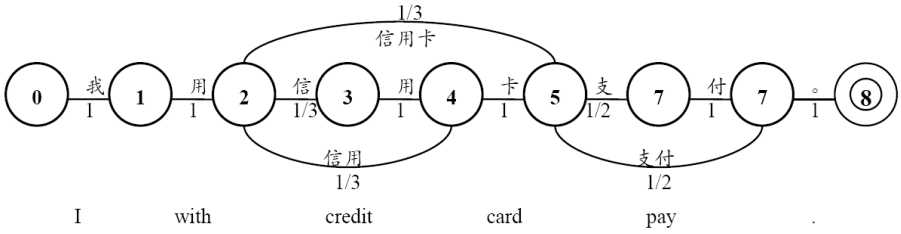

Figure 3: Example of a word lattice for a Chinese sentence

requires that the sentence pair meets the bilingual constraints. Such a segmentation process in the training stage facilitates the utilisation of word lattice decoding.

4.3.3 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, as in (2):

$$\hat{f}_1^K = \arg\max_{f_1^K, K} \{Pr(f_1^K | c_1^J)\} \tag{2}$$

Then in the decoding stage, we seek the translation of the most likely source segmentation as in (3):

$$\hat{e}_1^I = \arg\max_{e_1^I, I} \{Pr(e_1^I | \hat{f}_1^K)\} \tag{3}$$

In such a scenario, some segmentations which are potentially optimal for translation may be lost. This motivates the need for word lattice decoding. The search process can be rewritten as in (4)–(6):

$$\begin{aligned}
\hat{e}_1^I &= \arg\max_{e_1^I, I} \{\max_{f_1^K, K} Pr(e_1^I, f_1^K | c_1^J)\} && (4)\\
&= \arg\max_{e_1^I, I} \{\max_{f_1^K, K} Pr(e_1^I) Pr(f_1^K | e_1^I, c_1^J)\} && (5)\\
&\simeq \arg\max_{e_1^I, I} \{\max_{f_1^K, K} Pr(e_1^I) Pr(f_1^K | e_1^I) Pr(f_1^K | c_1^J)\} && (6)
\end{aligned}$$

Given the fact that the number of segmentations f_1^K grows exponentially with respect to the number of characters J, 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 which contains a very limited number of characters, making the utilisation of word lattices tractable in PB-SMT.

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 choose to extrinsically evaluate the performance of our approach on 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

For our word packing experiments, we used the Chinese–English datasets provided within the IWSLT 2006 and 2007 evaluation campaigns. This multilingual speech corpus contains sentences similar to those that are usually found in phrase-books for tourists going abroad Takezawa et al. [2002]. 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 IWSLT 2006 and 2007 test sets. We used both test sets because they are quite different in terms of sentence length and vocabulary size. Based on the original manual segmentation for Chinese, the various statistics for the corpora are shown in Table 6. To test the scalability of our approach, we added in the HIT corpus[13], which was made available for IWSLT 2008 evaluation campaign. This data set consists of around 120K sentence pairs in the dialogue domain.

For Chinese word segmentation experiments, in order to test the performance of our segmenter across different domains, we used data from parliamentary documents, i.e. a portion of UN data for the NIST 2006 evaluation campaign. The system was developed on the LDC Multiple-Translation Chinese (MTC) Corpus and tested on MTC part 2, which was also used as a test set for the NIST 2002 evaluation campaign. Based on Chinese segmentation using the Stanford segmenter, the various statistics for the UN corpora are shown in Table 7.

[13] http://mitlab.hit.edu.cn/index.php/resources/29-the-resource/111-share-bilingual-corpus.html

		Chinese		English
		Characters	Words	
Train	Sentences	40,958		
	Running words	488,303	357,968	385,065
	Vocabulary size	2,742	11,362	9,718
Dev.	Sentences	489 (7 ref.)		
	Running words	8,141	5,717	46,904
	Vocabulary size	835	1,143	1,786
Eval.	Sentences	489 (7 ref.)/489 (6 ref.)		
	Running words	8,793/4,377	6,066/3,166	51,500/23,181
	Vocabulary size	936/772	1,339/862	2,016/1,339

Table 6: Corpus statistics for the IWSLT task

		Chinese		English
		Characters	Words	
Train	Sentences	40,000		
	Running words	1,412,395	880,301	956,023
	Vocabulary size	6057	22,433	20,068
Dev.	Sentences	993 (9 ref.)		
	Running words	41,466	24,557	267,222
	Vocabulary size	1,983	4,931	10,665
Eval.	Sentences	878 (4 ref.)		
	Running words	38,700	22,652	105,530
	Vocabulary size	1,907	4,607	7,388

Table 7: Corpus statistics for the NIST task

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 Kneser and Ney [1995] 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 the single best segmentation and word lattices.

6 Experiments

6.1 Word Packing

6.1.1 Results

The initial word alignments are obtained using the baseline configuration described above. From these, we build two bilingual 1-to-n dictionaries (one for each direction), and the training corpus is updated by repacking the words in the

dictionaries, using the method presented in Section 3. As previously mentioned, this process can be repeated several times; at each step, we can also choose to exploit only one of the two available dictionaries, if so desired. 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 the features of the log-linear model and unpack the grouped words to recover the initial words. Finally, minimum error-rate training and decoding are performed.

	BLEU[%]	NIST	METEOR[%]
Baseline	**33.85**	6.3837	54.85
n=1. with C-E dict.	**35.02**	6.5145	55.55
n=1. with E-C dict.	34.83	6.4638	56.06
n=2. with C-E dict.	34.42	6.5553	55.74
n=2. with E-C dict.	**35.69**	6.6294	57.23

Table 8: Influence of word packing on the Chinese-to-English IWSLT 2007 task

The various parameters of the method (k, t_{COOC}, t_{AC}, cf. section 3.2) were optimised on the development set. We found out that it was enough to perform two iterations of repacking: the optimal set of values was found to be $k = 3$, $t_{AC} = 0.9$, $t_{COOC} = 20$ for packing English words and $t_{AC} = 0.3$, $t_{COOC} = 10$ for packing Chinese words in the first iteration, and $t_{AC} = 0.9$, $t_{COOC} = 8$ for packing English words and $t_{AC} = 0.7$, $t_{COOC} = 15$ for packing Chinese words in the second iteration.[14] In Table 8, we report the results obtained on the IWSLT 2007 test set, where n denotes the iteration. For each iteration, we first considered the inclusion of only the Chinese–English dictionary, and then only the English–Chinese dictionary.[15]

After the first step, we can already see an improvement over the baseline when considering one of the two dictionaries. More gain can be obtained by packing English words, leading to an increase of 1.17 absolute BLEU points (3.46% relative). The improvement is also confirmed by NIST and METEOR evaluation metrics. Moreover, we can gain from performing another step. However, the inclusion of the Chinese–English dictionary is harmful in this case, probably because 1-to-n alignments have been captured during the first step. By including the English–Chinese dictionary only, we can achieve an increase of 1.84 absolute BLEU points (5.44% relative) over the initial baseline, which is statistically significant ($p<0.01$).[16]

The improvement in performance can be attributed to better word alignment

[14] The parameters k, t_{AC}, and t_{COOC} are optimised for each step, and the alignment obtained using the best set of parameters for a given step are used as input for the following step.

[15] We intend to consider including both Chinese–English and English–Chinese dictionaries in future work. However, in this case the parameter optimisation is more complicated as we need to jointly optimise the parameters for both directions.

[16] Note that this setting (using only Chinese dictionary for the first step and only the English dictionary for the second step) is also the best setting on the development set.

after simplifying the alignment task after word packing, and subsequently higher quality phrasal translations for PB-SMT systems. Figure 4 gives two examples of better translation after word packing. Phrases such as "there 's", "will it take", and "get to" are packed words in the C-E bilingual dictionary so that valid phrase pairs can be included in the phrase table. Moreover, the probability of these valid phrase pairs can be boosted after word packing so that the correct hypothesis can survive in the decoding stage.

ZH: 在 巴黎 出 了 交通 事故 。
Baseline: in paris |0-1| out |2-3| a traffic accident |4-5| . |6-6|
WP: in paris |0-1| there 's |2-2| a traffic accident . |3-6|

ZH: 到 洛杉矶 需要 多 长 时间 ？
Baseline: how long |3-5| do i need |2-2| to los angeles |0-1| ? |6-6|
WP: how long will it take |3-5| to |2-2| get to los angeles |0-1| ? |6-6|

Figure 4: Translation examples using word packing

Quality of the Dictionaries To assess the quality of the extraction procedure, we simply manually evaluated the ratio of incorrect entries in the dictionaries. After one step of word packing, the Chinese–English and the English–Chinese dictionaries respectively contain 13.6% and 8.6% incorrect entries. After two steps of packing, they only contain 7.7% and 7.2% incorrect entries. More interestingly, some errors committed in the first step can be corrected in the second step, leading to a dictionary of higher quality. Some cases generally considered to be difficult such as m-to-n non-compositional phrasal alignments can also be identified in the second step.

6.1.2 Alignment Types

Intuitively, the word alignments obtained after word packing are more likely to be 1:1 than before. Indeed, the word sequences in one language that usually align to one single word in the other language have been grouped together to form one single token. Table 9 shows the detail of the distribution of alignment types after one and two steps of automatic repacking.

In particular, we can observe that the 1:1 alignments are more frequent after the application of repacking: the ratio of this type of alignment has increased by 8.2% for Chinese–English and 4.18% for English–Chinese.

6.1.3 Influence of Word Segmentation Approach

To test the influence of the initial word segmentation on the process of word packing, we considered an additional segmentation configuration, based on the LDC segmenter.

		1:0	1:1	1:2	1:3	1:n ($n>3$)
C-E	Base.	28.48	**57.08**	11.08	2.11	1.25
	n=1	27.30	57.68	11.49	2.19	1.32
	n=2	17.45	**65.28**	10.68	4.12	2.48
E-C	Base.	22.19	**59.60**	11.61	4.30	2.30
	n=1	21.27	62.91	9.82	3.74	2.25
	n=2	26.76	**63.78**	6.25	1.94	1.25

Table 9: Distribution of alignment types (%)

	BLEU[%]
Original segmentation	33.85
Original segmentation + Word packing	**35.02**
Automatic segmentation	31.74
Automatic segmentation + Word packing	**32.58**

Table 10: Influence of Chinese segmentation

The results obtained are displayed in Table 10. The automatic segmenter leads to lower results than the human-corrected segmentation. However, the proposed method seems to be beneficial irrespective of the choice of segmentation. Indeed, we can also observe an improvement in the new setting: 0.84 points absolute increase in BLEU (2.65% relative), which is statistically significant ($p<0.05$). The experimental results of word packing reported so far are based on either manual segmentation or automatic segmentation using monolingual segmenters. In next section, we show the results on directly using the word packing approach to perform word segmentation.

6.2 Word Segmentation

6.2.1 Results

The initial word alignments are obtained using the baseline configuration 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. To optimise the weights for the features of the log-linear model using minimum error-rate training, we segment the Chinese sentences in the development set using a simple dictionary-based maximum matching algorithm to obtain a single best segmentation.[17] 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. 4.2) were optimised on the development set. One iteration of

[17] In order to save computing time, we used the same set of parameters obtained above to decode both the single-best segmentation and the word lattice. Recent work has been done on lattice-based minimum error-rate training Macherey et al. [2008].

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 the 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 11, our bilingually motivated segmenter achieved statistically significantly ($p<0.03$) better results than character-based segmentation when enhanced with word lattice decoding.[18] 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 we are unable to obtain a high quality bilingual dictionary 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.[19] 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 the IWSLT data (cf. Tables 12 and 13), the improvements over character-based segmentation are both statistically significant ($p<0.03$ for IWSLT 2006 test set and $p<0.01$ for IWSLT 2007 test set respectively). Compared to the best in-domain segmenter, the LDC segmenter, our approach yielded a consistently good performance on both translation tasks. Moreover, the good performance is confirmed by all three evaluation measures.

From the experiments, we also observed that adding in word lattice machanism into the PB-SMT system trained on monolingually segmented data does not help or even harm the system due to the mismatch between PB-SMT training and decoding. Previous research also shows that combining phrase tables using different segmentations is necessary for word lattice decoding Dyer et al. [2008]. This is also the advantage of our bilingually motivated segmentation, which facilitates word lattice decoding because it can generate different segmentations for the same Chinese sentence given different target English translations.

6.2.2 Parameter Search Graph

The reliability estimation process is computationally intensive. However, this can easily be 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 5 shows 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.

[18] Note that 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.

[19] 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-

	BLEU[%]	NIST	METEOR[%]
CS	8.43	4.6272	37.78
Stanford	**10.45**	**5.0675**	36.99
Stanford-WordLattice	8.61	4.5456	37.15
BS-SingleBest	7.98	4.4374	35.10
BS-WordLattice	9.04	4.6667	**38.34**

Table 11: Bilingually motivated word segmentation on the NIST task

	BLEU[%]	NIST	METEOR[%]
CS	19.31	6.1816	49.98
LDC	20.37	6.2089	49.84
LDC-WordLattice	20.15	6.2876	50.51
BS-SingleBest	18.65	5.7816	46.02
BS-WordLattice	**20.41**	**6.2874**	**51.24**

Table 12: Bilingually motivated word segmentation on the IWSLT 2006 task

6.2.3 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 14 and 15 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 14, 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 (15.48%). 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.

6.2.4 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.

based segmentation.

	BLEU[%]	NIST	METEOR[%]
CS	29.59	6.1216	52.16
LDC	31.74	6.2464	54.03
LDC-WordLattice	**31.94**	6.2884	55.74
BS-SingleBest	30.23	6.0476	51.25
BS-WordLattice	31.71	**6.3518**	**56.03**

Table 13: Bilingually motivated word segmentation on the IWSLT 2007 task

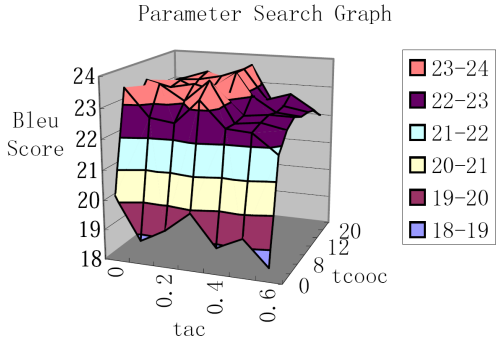

Figure 5: The search graph on the development set in the IWSLT task

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.

As we can see from Table 16, our segmentation approach achieved consistent results on both the IWSLT 2006 and 2007 test sets. On the NIST task (cf. Table 17), 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 6.2.1).

6.2.5 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 18 shows the results obtained on the IWSLT data set using the MTTK alignment tool Deng and Byrne [2005, 2006].

7 Related Work

Fertility-based models such as IBM models 3, 4, and 5 allow for alignments between one word and several words in order to capture the pervasive 1-to-n

	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 14: Vocabulary size of the NIST task (40K)

	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 15: Vocabulary size of the IWSLT task (40K)

correspondences. They can be seen as extensions of the simpler IBM models 1 and 2 Brown et al. [1993]. Similarly, Deng and Byrne Deng and Byrne [2005] proposed an HMM framework with special attention to dealing with 1-to-n alignment, which is an extension of the original model of Vogel et al. [1996]. However, as mentioned above, these models rarely question the monolingual tokenisation, i.e. the basic unit of the alignment process is the word. One alternative to extending the expressivity of one model (and usually its complexity) is to focus on the *input representation*; in particular, we argue that the alignment process can benefit from a simplification of the input, which consists of trying to reduce the number of 1-to-n alignments to consider. Note that the need to consider segmentation and alignment at the same time is also mentioned in Tiedemann [2003], and related issues are reported in Wu [1997].

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 PB-SMT to achieve competitive results compared to using monolingual segmenters. However, Xu et al. [2004] used word aligners to reconstruct a (monolingual) Chinese dictionary and reuse this dictionary to segment Chinese sentences as other monolingual segmenters do. Our approach features the use of a bilingual dictionary and conducts segmentation based on the bilingual dictionary. 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 are based on a poor phrase-based SMT baseline and the issue of scalability is not addressed.

Xu et al. [2005] were the first to propose the use of word lattice decoding in PB-SMT, in order to address the problems that segmentation posed on the decoding. Dyer et al. [2008] extended this approach to hierarchical SMT systems

	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 16: Scaling up to 160K on IWSLT data sets (BLEU[%])

	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 17: Scalability of bilingually motivated word segmentation on the NIST task (BLEU[%])

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, making the training and decoding processes more coherent. More recently, Xu et al. [2008] proposed a Bayesian semi-supervised model for word segmentation by combining knowledge from both monolingual segmentation and bilingual word alignment. Our approach is not specifically designed for segmentation; it is a new mechanism that can automatically perform bidirectional segmentation optimisation. The boostrapping step can help the statistical aligners overcome the limitations posed by the first-order assumption. The bidirectional word packing can also overcome the shortcomings of the 1-to-n assumption inherent in IBM models by facilitating m-to-n alignment structures (cf. Fraser and Marcu [2007]). On the other hand, our approach can be generalised to perform word segmentation without relying on any monolingual resources, such as dictionaries and the like.

8 Conclusions and Future Work

In this paper, we have introduced a simple yet effective method to pack words together in order to give a different and simplified input to automatic word aligners. We use a bootstrapping approach in which we first extract 1-to-n word alignments using an existing word aligner, and then estimate the confidence of those alignments to decide whether or not the n words have to be grouped; if so, this group is considered a new basic unit to consider. We can

	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 18: Bilingually motivated word segmentation on IWSLT data sets using MTTK (BLEU[%])

finally reapply the word aligner on the updated sentences. This approach can be used for bootstrapping word alignments based on any monolingual word segmentation; it can also be used for direct word segmentation without relying on any monolingual segmenters.

We have evaluated the performance of our approach by measuring the influence of this process on the Chinese–English MT task based on the IWSLT 2007 evaluation campaign with a reasonally small amount of training data. We report a 1.84 points absolute (5.44% relative) increase in BLEU score over a standard phrase-based SMT system. We have verified that this process actually reduces the number of 1-to-n alignments with $n \neq 1$, and that it is independent of the (Chinese) segmentation strategy. We then generalise our approach for direct Chinese word segmentation without relying on monolingual word segmenters and demonstrate that (i) our approach is not as sensitive to the domain as monolingual segmenters, and (ii) the SMT system using our word segmentation can achieve state-of-the-art performance. Moreover, our approach can be scaled up to larger data sets and achieves competitive results if the small data used is a representative sample of the larger one. Since our approach does not rely on monolingual segmenters, it is particularly useful for languages which lack manually segmented resources in the context of SMT.

Our approach is also faced with a few limitations. First of all, such an approach is built on existing fertility-based word alignment models which are computationally expensive in the process of parameter search. Despite the fact that it is very effective when used on a relatively small data set, using a small data set tends to suffer from data sparseness problems especially when a large vocabulary exists.[20] Therefore, a successful scaling up of our method has to meet two constraints: (i) the data has a relatively small vocabulary so that a high-quality bilingual dictionary can be obtained; and (ii) the small data set is representative enough for the larger data set. Given such limitations, in future work, we firstly intend to explore the correlation between vocabulary size and the amount of training data needed in order to achieve good results. We also plan to use more sophisticated association measures to estimate the reliability of the derived 1-to-n alignments. Then, we will take a further step

[20]From our experiments, word packing cannot effectively improve PB-SMT when using a small set of UN data simply because this data contains a large vocabulary.

to derive 1-to-n alignments using syntactic information and heuristics Melamed [1997] rather than existing IBM models and integrate such information into probabilistic word aligners. By doing so, we can overcome both limitations. We also plan to conduct experiments using larger data sets and more domains, including newswire and controlled technical corpus for localisation purposes.

Acknowledgments

This work is supported by Science Foundation Ireland[21] (O5/IN/1732) and the Irish Centre for High-End Computing.[22] We would like to thank the reviewers for their critical and insightful comments.

References

Satanjeev Banerjee and Alon Lavie. 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, Michigan, 2005.

Alexandra Birch, Chris Callison-Burch, and Miles Osborne. Constraining the phrase-based, joint probability statistical translation model. In *Proceedings of the 7th biennial conference of the Association for Machine Translation in the Americas*, pages 10–18, Boston, MA, 2006.

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

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

Yonggang Deng and William Byrne. 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, Canada, 2005.

Yonggang Deng and William Byrne. 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, 2006.

[21] `http://www.sfi.ie/`
[22] `http://www.ichec.ie/`

George Doddington. 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, 2002.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 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, 2008.

Alexander Fraser and Daniel Marcu. Getting the structure right for word alignment: LEAF. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 51–60, Prague, Czech Republic, 2007.

Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 181–184, Detroit, MI, 1995.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X*, pages 79–86, Phuket, Thailand, 2005.

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

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. 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, 2007.

Yanjun Ma, Nicolas Stroppa, and Andy Way. 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, 2007.

Yanjun Ma, Sylwia Ozdowska, Yanli Sun, and Andy Way. Improving word alignment using syntactic dependencies. In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 69–77, Columbus, OH, 2008.

Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 725–734, Honolulu, HI, 2008.

Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 133–139, Morristown, NJ, 2002.

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

I. Dan Melamed. Automatic discovery of non-compositional compounds in parallel data. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 97–108, Somerset, NJ, 1997.

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

Franz Och. 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, 2003.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

Michael Paul. Overview of the IWSLT 2006 Evaluation Campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 1–15, Kyoto, Japan, 2006.

Frank Smadja, Kathleen R. McKeown, and Vasileios Hatzivassiloglou. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1):1–38, 1996.

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

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

Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proceedings of Third International Conference on Language Resources and Evaluation 2002*, pages 147–152, Las Palmas, Canary Islands, Spain, 2002.

Jörg Tiedemann. Combining clues for word alignment. In *Proceedings of the 10th Conference of European Chapter of the Association for Computational Linguistics*, pages 339–346, Budapest, Hungary, 2003.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 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, 2005.

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

Dekai Wu. Stochastic Inversion Transduction Grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, 1997.

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

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

Jia Xu, Jianfeng Gao, Kristina Toutanova, and Hermann Ney. Bayesian semi-supervised chinese word segmentation for statistical machine translation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1017–1024, Manchester, UK, 2008.

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

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