Bilingual Sense Similarity for Statistical Machine Translation

Boxing Chen, George Foster and **Roland Kuhn**
National Research Council Canada
283 Alexandre-Taché Boulevard, Gatineau (Québec), Canada J8X 3X7
{Boxing.Chen, George.Foster, Roland.Kuhn}@nrc.ca

Abstract

This paper proposes new algorithms to compute the sense similarity between two units (words, phrases, rules, *etc.*) from parallel corpora. The sense similarity scores are computed by using the vector space model. We then apply the algorithms to statistical machine translation by computing the sense similarity between the source and target side of translation rule pairs. Similarity scores are used as additional features of the translation model to improve translation performance. Significant improvements are obtained over a state-of-the-art hierarchical phrase-based machine translation system.

1 Introduction

The sense of a term can generally be inferred from its context. The underlying idea is that a term is characterized by the contexts it co-occurs with. This is also well known as *the Distributional Hypothesis* (Harris, 1954): terms occurring in similar contexts tend to have similar meanings. There has been a lot of work to compute the sense similarity between terms based on their distribution in a corpus, such as (Hindle, 1990; Lund and Burgess, 1996; Landauer and Dumais, 1997; Lin, 1998; Turney, 2001; Pantel and Lin, 2002; Pado and Lapata, 2007).

In the work just cited, a common procedure is followed. Given two terms to be compared, one first extracts various features for each term from their contexts in a corpus and forms a vector space model (VSM); then, one computes their similarity by using similarity functions. The features include words within a surface window of a fixed size (Lund and Burgess, 1996), grammatical dependencies (Lin, 1998; Pantel and Lin 2002; Pado and Lapata, 2007), *etc.* The similarity function which has been most widely used is cosine distance (Salton and McGill, 1983); other similarity functions include Euclidean distance, City Block distance (Bullinaria and Levy; 2007), and Dice and Jaccard coefficients (Frakes and Baeza-Yates, 1992), *etc.* Measures of monolingual sense similarity have been widely used in many applications, such as synonym recognizing (Landauer and Dumais, 1997), word clustering (Pantel and Lin 2002), word sense disambiguation (Yuret and Yatbaz 2009), *etc.*

Use of the vector space model to compute sense similarity has also been adapted to the multilingual condition, based on the assumption that two terms with similar meanings often occur in comparable contexts across languages. Fung (1998) and Rapp (1999) adopted VSM for the application of extracting translation pairs from comparable or even unrelated corpora. The vectors in different languages are first mapped to a common space using an initial bilingual dictionary, and then compared.

However, there is no previous work that uses the VSM to compute sense similarity for terms from parallel corpora. The sense similarities, i.e. the translation probabilities in a translation model, for units from parallel corpora are mainly based on the co-occurrence counts of the two units. Therefore, questions emerge: how good is the sense similarity computed via VSM for two units from parallel corpora? Is it useful for multilingual applications, such as statistical machine translation (SMT)?

In this paper, we try to answer these questions, focusing on sense similarity applied to the SMT task. For this task, translation rules are heuristically extracted from automatically word-aligned sentence pairs. Due to noise in the training corpus or wrong word alignment, the source and target sides of some rules are not semantically equivalent, as can be seen from the following

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 834–843,
Uppsala, Sweden, 11-16 July 2010.

real examples which are taken from the rule table built on our training data (Section 5.1):

世界 上 X 之一 ||| *one of X* (*)
世界 上 X 之一 ||| *one of X in the world*
许多 市民 ||| *many citizens*
许多 市民 ||| *many hong kong residents* (*)

The source and target sides of the rules with (*) at the end are not semantically equivalent; it seems likely that measuring the semantic similarity from their context between the source and target sides of rules might be helpful to machine translation.

In this work, we first propose new algorithms to compute the sense similarity between two units (unit here includes word, phrase, rule, *etc*.) in different languages by using their contexts. Second, we use the sense similarities between the source and target sides of a translation rule to improve statistical machine translation performance.

This work attempts to measure directly the sense similarity for units from different languages by comparing their contexts[1]. Our contribution includes proposing new bilingual sense similarity algorithms and applying them to machine translation.

We chose a hierarchical phrase-based SMT system as our baseline; thus, the units involved in computation of sense similarities are hierarchical rules.

2 Hierarchical phrase-based MT system

The hierarchical phrase-based translation method (Chiang, 2005; Chiang, 2007) is a formal syntax-based translation modeling method; its translation model is a weighted synchronous context free grammar (SCFG). No explicit linguistic syntactic information appears in the model. An SCFG rule has the following form:

$$X \rightarrow \langle \alpha, \gamma, \sim \rangle$$

where X is a non-terminal symbol shared by all the rules; each rule has at most two non-terminals. α (γ) is a source (target) string consisting of terminal and non-terminal symbols. $\sim$ defines a one-to-one correspondence between non-terminals in α and γ.

[1] There has been a lot of work (more details in Section 7) on applying word sense disambiguation (WSD) techniques in SMT for translation selection. However, WSD techniques for SMT do so indirectly, using source-side context to help select a particular translation for a source rule.

	source	target
Ini. phr.	*他 出席 了 会议*	*he attended the meeting*
Rule 1 Context 1	*他 出席 了 X_1* *会议*	*he attended X_1* *the, meeting*
Rule 2 Context 2	*会议* *他, 出席, 了*	*the meeting* *he, attended*
Rule 3 Context 3	*他 X_1 会议* *出席, 了*	*he X_1 the meeting* *attended*
Rule 4 Context 4	*出席 了* *他,会议*	*attended* *he, the, meeting*

Figure 1: example of hierarchical rule pairs and their context features.

Rule frequencies are counted during rule extraction over word-aligned sentence pairs, and they are normalized to estimate features on rules. Following (Chiang, 2005; Chiang, 2007), 4 features are computed for each rule:

- $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$ are direct and inverse rule-based conditional probabilities;
- $P_w(\gamma|\alpha)$ and $P_w(\alpha|\gamma)$ are direct and inverse lexical weights (Koehn et al., 2003).

Empirically, this method has yielded better performance on language pairs such as Chinese-English than the phrase-based method because it permits phrases with gaps; it generalizes the normal phrase-based models in a way that allows long-distance reordering (Chiang, 2005; Chiang, 2007). We use the *Joshua* implementation of the method for decoding (Li et al., 2009).

3 Bag-of-Words Vector Space Model

To compute the sense similarity via VSM, we follow the previous work (Lin, 1998) and represent the source and target side of a rule by feature vectors. In our work, each feature corresponds to a context word which co-occurs with the translation rule.

3.1 Context Features

In the hierarchical phrase-based translation method, the translation rules are extracted by abstracting some words from an initial phrase pair (Chiang, 2005). Consider a rule with non-terminals on the source and target side; for a given instance of the rule (a particular phrase pair in the training corpus), the context will be the words instantiating the non-terminals. In turn, the context for the sub-phrases that instantiate the non-terminals will be the words in the remainder of the phrase pair. For example in Figure 1, if we

have an initial phrase pair *他 出席 了 会议 ||| he attended the meeting*, and we extract four rules from this initial phrase: *他 出席 了 X_1 ||| he attended X_1*, *会议 ||| the meeting*, *他 X_1 会议 ||| he X_1 the meeting*, and *出席 了 ||| attended*. Therefore, *the* and *meeting* are context features of target pattern *he attended X_1*; *he* and *attended* are the context features of *the meeting*; *attended* is the context feature of *he X_1 the meeting*; also *he, the* and *meeting* are the context feature of *attended* (in each case, there are also source-side context features).

3.2 Bag-of-Words Model

For each side of a translation rule pair, its context words are all collected from the training data, and two "bags-of-words" which consist of collections of source and target context words co-occurring with the rule's source and target sides are created.

$$B_f = \{f_1, f_2, ..., f_I\}$$
$$B_e = \{e_1, e_2, ..., e_J\} \quad (1)$$

where $f_i (1 \le i \le I)$ are source context words which co-occur with the source side of rule α, and $e_j (1 \le j \le J)$ are target context words which co-occur with the target side of rule γ.

Therefore, we can represent source and target sides of the rule by vectors $\vec{v}_f$ and $\vec{v}_e$ as in Equation (2):

$$\vec{v}_f = \{w_{f_1}, w_{f_2}, ..., w_{f_I}\}$$
$$\vec{v}_e = \{w_{e_1}, w_{e_2}, ..., w_{e_J}\} \quad (2)$$

where w_{f_i} and w_{e_j} are values for each source and target context feature; normally, these values are based on the counts of the words in the corresponding bags.

3.3 Feature Weighting Schemes

We use pointwise mutual information (Church et al., 1990) to compute the feature values. Let c ($c \in B_f$ or $c \in B_e$) be a context word and $F(r,c)$ be the frequency count of a rule r (α or γ) co-occurring with the context word c. The pointwise mutual information $MI(r,c)$ is defined as:

$$w(r,c) = MI(r,c) = \frac{\log \frac{F(r,c)}{N}}{\log \frac{F(r)}{N} \times \log \frac{F(c)}{N}} \quad (3)$$

where N is the total frequency counts of all rules and their context words. Since we are using this value as a weight, following (Turney, 2001), we drop *log, N* and $F(r)$. Thus (3) simplifies to:

$$w(r,c) = \frac{F(r,c)}{F(c)} \quad (4)$$

It can be seen as an estimate of $P(r \mid c)$, the empirical probability of observing r given c.

A problem with $P(r \mid c)$ is that it is biased towards infrequent words/features. We therefore smooth $w(r,c)$ with *add-k* smoothing:

$$w(r,c) = \frac{F(r,c)+k}{\sum_{i=1}^{R}(F(r_i,c)+k)} = \frac{F(r,c)+k}{F(c)+kR} \quad (5)$$

where k is a tunable global smoothing constant, and R is the number of rules.

4 Similarity Functions

There are many possibilities for calculating similarities between bags-of-words in different languages. We consider IBM model 1 probabilities and cosine distance similarity functions.

4.1 IBM Model 1 Probabilities

For the IBM model 1 similarity function, we take the geometric mean of symmetrized conditional IBM model 1 (Brown et al., 1993) bag probabilities, as in Equation (6).

$$sim(\alpha, \gamma) = sqrt(P(B_f \mid B_e) \cdot P(B_e \mid B_f)) \quad (6)$$

To compute $P(B_f \mid B_e)$, IBM model 1 assumes that all source words are conditionally independent, so that:

$$P(B_f \mid B_e) = \prod_{i=1}^{I} p(f_i \mid B_e) \quad (7)$$

To compute, we use a "Noisy-OR" combination which has shown better performance than standard IBM model 1 probability, as described in (Zens and Ney, 2004):

$$p(f_i \mid B_e) = 1 - p(\bar{f}_i \mid B_e) \quad (8)$$

$$p(f_i \mid B_e) \approx 1 - \prod_{j=1}^{J}(1 - p(f_i \mid e_j)) \quad (9)$$

where $p(\bar{f}_i \mid B_e)$ is the probability that f_i is *not* in the translation of B_e, and is the IBM model 1 probability.

4.2 Vector Space Mapping

A common way to calculate semantic similarity is by vector space cosine distance; we will also

use this similarity function in our algorithm. However, the two vectors in Equation (2) cannot be directly compared because the axes of their spaces represent different words in different languages, and also their dimensions *I* and *J* are not assured to be the same. Therefore, we need to first map a vector into the space of the other vector, so that the similarity can be calculated. Fung (1998) and Rapp (1999) map the vector one-dimension-to-one-dimension (a context word is a dimension in each vector space) from one language to another language via an initial bilingual dictionary. We follow (Zhao et al., 2004) to do vector space mapping.

Our goal is – given a source pattern – to distinguish between the senses of its associated target patterns. Therefore, we map all vectors in target language into the vector space in the source language. What we want is a representation $\vec{v}_a$ in the source language space of the target vector $\vec{v}_e$. To get $\vec{v}_a$, we can let $w_a^{f_i}$, the weight of the i^{th} source feature, be a linear combination over target features. That is to say, given a source feature weight for f_i, each target feature weight is linked to it with some probability. So that we can calculate a transformed vector from the target vectors by calculating weights $w_a^{f_i}$ using a translation lexicon:

$$w_a^{f_i} = \sum_{j=1}^{J} \Pr(f_i \mid e_j) w_{e_j} \tag{10}$$

where $p(f_i \mid e_j)$ is a lexical probability (we use IBM model 1 probability). Now the source vector and the mapped vector $\vec{v}_a$ have the same dimensions as shown in (11):

$$\begin{aligned} \vec{v}_f &= \{w_{f_1}, w_{f_2}, ..., w_{f_I}\} \\ \vec{v}_a &= \{w_a^{f_1}, w_a^{f_2}, ..., w_a^{f_I}\} \end{aligned} \tag{11}$$

4.3 Naïve Cosine Distance Similarity

The standard cosine distance is defined as the inner product of the two vectors $\vec{v}_f$ and $\vec{v}_a$ normalized by their norms. Based on Equation (10) and (11), it is easy to derive the similarity as follows:

$$\begin{aligned} sim(\alpha,\gamma) &= \cos(\vec{v}_f, \vec{v}_a) = \frac{\vec{v}_f \cdot \vec{v}_a}{|\vec{v}_f| \cdot |\vec{v}_a|} \\ &= \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} w_{f_i} \Pr(f_i \mid e_j) w_{e_j}}{sqrt(\sum_{I=1}^{I} w_{f_i}^2) sqrt(\sum_{i=1}^{I} {w_a^{f_i}}^2)} \end{aligned} \tag{12}$$

where *I* and *J* are the number of the words in source and target bag-of-words; w_{f_i} and w_{e_j} are values of source and target features; $w_a^{f_i}$ is the transformed weight mapped from all target features to the source dimension at word f_i.

4.4 Improved Similarity Function

To incorporate more information than the original similarity functions – IBM model 1 probabilities in Equation (6) and naïve cosine distance similarity function in Equation (12) – we refine the similarity function and propose a new algorithm.

As shown in Figure 2, suppose that we have a rule pair (α, γ). C_f^{full} and C_e^{full} are the contexts extracted according to the definition in section 3 from the full training data for α and for γ, respectively. C_f^{cooc} and C_e^{cooc} are the contexts for α and γ when α and γ co-occur. Obviously, they satisfy the constraints: $C_f^{cooc} \subseteq C_f^{full}$ and $C_e^{cooc} \subseteq C_e^{full}$. Therefore, the original similarity functions are to compare the two context vectors built on full training data directly, as shown in Equation (13).

$$sim(\alpha,\gamma) = sim(C_f^{full}, C_e^{full}) \tag{13}$$

Then, we propose a new similarity function as follows:

$$\begin{aligned} &sim(\alpha,\gamma) = \\ &sim(C_f^{full}, C_f^{cooc})^{\lambda_1} \cdot sim(C_f^{cooc}, C_e^{cooc})^{\lambda_2} \cdot sim(C_e^{full}, C_e^{cooc})^{\lambda_3} \end{aligned} \tag{14}$$

where the parameters λ_i (i=1,2,3) can be tuned via minimal error rate training (MERT) (Och, 2003).

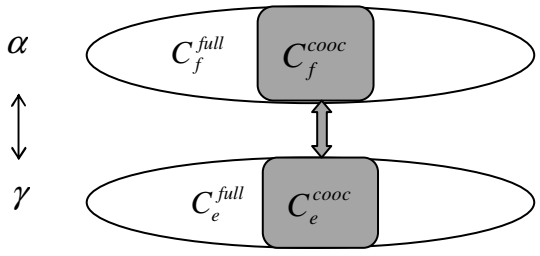

Figure 2: contexts for rule α and γ.

A unit's sense is defined by all its contexts in the whole training data; it may have a lot of different senses in the whole training data. However, when it is linked with another unit in the other language, its sense pool is constrained and is just

a subset of the whole sense set. $sim(C_f^{full}, C_f^{cooc})$ is the metric which evaluates the similarity between the whole sense pool of α and the sense pool when α co-occurs with γ; $sim(C_e^{full}, C_e^{cooc})$ is the analogous similarity metric for γ. They range from 0 to 1. These two metrics both evaluate the similarity for two vectors in the same language, so using cosine distance to compute the similarity is straightforward. And we can set a relatively large size for the vector, since it is not necessary to do vector mapping as the vectors are in the same language. $sim(C_f^{cooc}, C_e^{cooc})$ computes the similarity between the context vectors when α and γ co-occur. We may compute $sim(C_f^{cooc}, C_e^{cooc})$ by using IBM model 1 probability and cosine distance similarity functions as Equation (6) and (12). Therefore, on top of the degree of bilingual semantic similarity between a source and a target translation unit, we have also incorporated the monolingual semantic similarity between all occurrences of a source or target unit, and that unit's occurrence as part of the given rule, into the sense similarity measure.

5 Experiments

We evaluate the algorithm of bilingual sense similarity via machine translation. The sense similarity scores are used as feature functions in the translation model.

5.1 Data

We evaluated with different language pairs: Chinese-to-English, and German-to-English. For Chinese-to-English tasks, we carried out the experiments in two data conditions. The first one is the *large data* condition, based on training data for the NIST [2] 2009 evaluation Chinese-to-English track. In particular, all the allowed bilingual corpora except the *UN corpus* and *Hong Kong Hansard corpus* have been used for estimating the translation model. The second one is the *small data* condition where only the *FBIS*[3] corpus is used to train the translation model. We trained two language models: the first one is a 4-gram LM which is estimated on the target side of the texts used in the *large data* condition. The second LM is a 5-gram LM trained on the so-called English *Gigaword corpus*. Both language models are used for both tasks.

We carried out experiments for translating Chinese to English. We use the same development and test sets for the two data conditions. We first created a development set which used mainly data from the NIST 2005 test set, and also some balanced-genre web-text from the NIST training material. Evaluation was performed on the NIST 2006 and 2008 test sets. Table 1 gives figures for training, development and test corpora; |S| is the number of the sentences, and |W| is the number of running words. Four references are provided for all dev and test sets.

			Chi	Eng
Parallel Train	Large Data	|S|	3,322K	
		|W|	64.2M	62.6M
	Small Data	|S|	245K	
		|W|	9.0M	10.5M
Dev		|S|	1,506	1,506×4
Test	NIST06	|S|	1,664	1,664×4
	NIST08	|S|	1,357	1,357×4
Gigaword		|S|	-	11.7M

Table 1: Statistics of training, dev, and test sets for Chinese-to-English task.

For German-to-English tasks, we used WMT 2006[4] data sets. The parallel training data contains 21 million target words; both the dev set and test set contain 2000 sentences; one reference is provided for each source input sentence. Only the target-language half of the parallel training data are used to train the language model in this task.

5.2 Results

For the baseline, we train the translation model by following (Chiang, 2005; Chiang, 2007) and our decoder is *Joshua*[5], an open-source hierarchical phrase-based machine translation system written in Java. Our evaluation metric is IBM BLEU (Papineni et al., 2002), which performs case-insensitive matching of *n*-grams up to $n = 4$. Following (Koehn, 2004), we use the bootstrap-resampling test to do significance testing.

By observing the results on dev set in the additional experiments, we first set the smoothing constant k in Equation (5) to 0.5.

Then, we need to set the sizes of the vectors to balance the computing time and translation accu-

[2] http://www.nist.gov/speech/tests/mt
[3] LDC2003E14
[4] http://www.statmt.org/wmt06/
[5] http://www.cs.jhu.edu/~ccb/joshua/index.html

racy, *i.e.*, we keep only the top N context words with the highest feature value for each side of a rule [6]. In the following, we use "Alg1" to represent the original similarity functions which compare the two context vectors built on full training data, as in Equation (13); while we use "Alg2" to represent the improved similarity as in Equation (14). "IBM" represents IBM model 1 probabilities, and "COS" represents cosine distance similarity function.

After carrying out a series of additional experiments on the *small data* condition and observing the results on the dev set, we set the size of the vector to 500 for Alg1; while for Alg2, we set the sizes of C_f^{full} and C_e^{full} N_1 to 1000, and the sizes of C_f^{cooc} and C_e^{cooc} N_2 to 100.

The sizes of the vectors in Alg2 are set in the following process: first, we set N_2 to 500 and let N_1 range from 500 to 3,000, we observed that the dev set got best performance when N_1 was 1000; then we set N_1 to 1000 and let N_1 range from 50 to 1000, we got best performance when N_1=100. We use this setting as the default setting in all remaining experiments.

Algorithm	NIST'06	NIST'08
Baseline	27.4	21.2
Alg1 IBM	27.8*	21.5
Alg1 COS	27.8*	21.5
Alg2 IBM	27.9*	21.6*
Alg2 COS	**28.1****	**21.7***

Table 2: Results (BLEU%) of *small data* Chinese-to-English NIST task. Alg1 represents the original similarity functions as in Equation (13); while Alg2 represents the improved similarity as in Equation (14). IBM represents IBM model 1 probability, and COS represents cosine distance similarity function. * or ** means result is significantly better than the baseline ($p < 0.05$ or $p < 0.01$, respectively).

	Ch-En		De-En
Algorithm	NIST'06	NIST'08	Test'06
Baseline	31.0	23.8	26.9
Alg2 IBM	31.5*	24.5**	27.2*
Alg2 COS	**31.6****	**24.5****	**27.3***

Table 3: Results (BLEU%) of *large data* Chinese-to-English NIST task and German-to-English WMT task.

[6] We have also conducted additional experiments by removing the stop words from the context vectors; however, we did not observe any consistent improvement. So we filter the context vectors by only considering the feature values.

Table 2 compares the performance of Alg1 and Alg2 on the Chinese-to-English *small data* condition. Both Alg1 and Alg2 improved the performance over the baseline, and Alg2 obtained slight and consistent improvements over Alg1. The improved similarity function Alg2 makes it possible to incorporate monolingual semantic similarity on top of the bilingual semantic similarity, thus it may improve the accuracy of the similarity estimate. Alg2 significantly improved the performance over the baseline. The Alg2 cosine similarity function got 0.7 BLEU-score (p<0.01) improvement over the baseline for NIST 2006 test set, and a 0.5 BLEU-score (p<0.05) for NIST 2008 test set.

Table 3 reports the performance of Alg2 on Chinese-to-English NIST *large data* condition and German-to-English WMT task. We can see that IBM model 1 and cosine distance similarity function both obtained significant improvement on all test sets of the two tasks. The two similarity functions obtained comparable results.

6 Analysis and Discussion

6.1 Effect of Single Features

In Alg2, the similarity score consists of three parts as in Equation (14): $sim(C_f^{full}, C_f^{cooc})$, $sim(C_e^{full}, C_e^{cooc})$, and $sim(C_f^{cooc}, C_e^{cooc})$; where $sim(C_f^{cooc}, C_e^{cooc})$ could be computed by IBM model 1 probabilities $sim_{IBM}(C_f^{cooc}, C_e^{cooc})$ or cosine distance similarity function $sim_{COS}(C_f^{cooc}, C_e^{cooc})$. Therefore, our first study is to determine which one of the above four features has the most impact on the result. Table 4 shows the results obtained by using each of the 4 features. First, we can see that $sim_{IBM}(C_f^{cooc}, C_e^{cooc})$ always gives a better improvement than $sim_{COS}(C_f^{cooc}, C_e^{cooc})$. This is because $sim_{IBM}(C_f^{cooc}, C_e^{cooc})$ scores are more diverse than the latter when the number of context features is small (there are many rules that have only a few contexts.) For an extreme example, suppose that there is only one context word in each vector of source and target context features, and the translation probability of the two context words is not 0. In this case, $sim_{IBM}(C_f^{cooc}, C_e^{cooc})$ reflects the translation probability of the context word pair, while $sim_{COS}(C_f^{cooc}, C_e^{cooc})$ is always 1.

Second, $sim(C_f^{full}, C_f^{cooc})$ and $sim(C_e^{full}, C_e^{cooc})$ also give some improvements even when used

independently. For a possible explanation, consider the following example. The Chinese word "红" can translate to "red", "communist", or "hong" (the transliteration of 红, when it is used in a person's name). Since these translations are likely to be associated with very different source contexts, each will have a low $sim(C_f^{full}, C_f^{cooc})$ score. Another Chinese word 小溪 may translate into synonymous words, such as "brook", "stream", and "rivulet", each of which will have a high $sim(C_f^{full}, C_f^{cooc})$ score. Clearly, 红 is a more "dangerous" word than 小溪, since choosing the wrong translation for it would be a bad mistake. But if the two words have similar translation distributions, the system cannot distinguish between them. The monolingual similarity scores give it the ability to avoid "dangerous" words, and choose alternatives (such as larger phrase translations) when available.

Third, the similarity function of Alg2 consistently achieved further improvement by incorporating the monolingual similarities computed for the source and target side. This confirms the effectiveness of our algorithm.

	CE_LD		CE_SD	
testset (NIST)	'06	'08	'06	'08
Baseline	31.0	23.8	27.4	21.2
$sim(C_f^{full}, C_f^{cooc})$	31.1	24.3	27.5	21.3
$sim(C_e^{full}, C_e^{cooc})$	31.1	23.9	27.9	21.5
$sim_{IBM}(C_f^{cooc}, C_e^{cooc})$	31.4	24.3	27.9	21.5
$sim_{COS}(C_f^{cooc}, C_e^{cooc})$	31.2	23.9	27.7	21.4
Alg2 IBM	31.5	24.5	27.9	21.6
Alg2 COS	**31.6**	**24.5**	**28.1**	**21.7**

Table 4: Results (BLEU%) of Chinese-to-English *large data* (CE_LD) and *small data* (CE_SD) NIST task by applying one feature.

6.2 Effect of Combining the Two Similarities

We then combine the two similarity scores by using both of them as features to see if we could obtain further improvement. In practice, we use the four features in Table 4 together.

Table 5 reports the results on the *small data* condition. We observed further improvement on dev set, but failed to get the same improvements on test sets or even lost performance. Since the IBM+COS configuration has one extra feature, it is possible that it overfits the dev set.

Algorithm	Dev	NIST'06	NIST'08
Baseline	20.2	27.4	21.2
Alg2 IBM	20.5	27.9	21.6
Alg2 COS	20.6	28.1	21.7
Alg2 IBM+COS	20.8	27.9	21.5

Table 5: Results (BLEU%) for combination of two similarity scores. Further improvement was only obtained on dev set but not on test sets.

6.3 Comparison with Simple Contextual Features

Now, we try to answer the question: can the similarity features computed by the function in Equation (14) be replaced with some other simple features? We did additional experiments on *small data* Chinese-to-English task to test the following features: (15) and (16) represent the sum of the counts of the context words in C^{full}, while (17) represents the proportion of words in the context of α that appeared in the context of the rule (α, γ); similarly, (18) is related to the properties of the words in the context of γ.

$$N_f(\alpha) = \sum_{f_i \in C_f^{full}} F(\alpha, f_i) \quad (15)$$

$$N_e(\gamma) = \sum_{e_j \in C_e^{full}} F(\gamma, e_j) \quad (16)$$

$$E_f(\alpha, \gamma) = \frac{\sum_{f_i \in C_f^{cooc}} F(\alpha, f_i)}{N_f(\alpha)} \quad (17)$$

$$E_e(\alpha, \gamma) = \frac{\sum_{e_j \in C_e^{cooc}} F(\gamma, e_j)}{N_e(\gamma)} \quad (18)$$

where $F(\alpha, f_i)$ and $F(\gamma, e_j)$ are the frequency counts of rule α or γ co-occurring with the context word f_i or e_j respectively.

Feature	Dev	NIST'06	NIST'08
Baseline	20.2	27.4	21.2
$+N_f$	20.5	27.6	21.4
$+N_e$	20.5	27.5	21.3
$+E_f$	20.4	27.5	21.2
$+E_e$	20.4	27.3	21.2
$+N_f+N_e$	20.5	27.5	21.3

Table 6: Results (BLEU%) of using simple features based on context on *small data* NIST task. Some improvements are obtained on dev set, but there was no significant effect on the test sets.

Table 6 shows results obtained by adding the above features to the system for the *small data*

condition. Although all these features have obtained some improvements on dev set, there was no significant effect on the test sets. This means simple features based on context, such as the sum of the counts of the context features, are not as helpful as the sense similarity computed by Equation (14).

6.4 Null Context Feature

There are two cases where no context word can be extracted according to the definition of context in Section 3.1. The first case is when a rule pair is always a full sentence-pair in the training data. The second case is when for some rule pairs, either their source or target contexts are out of the span limit of the initial phrase, so that we cannot extract contexts for those rule-pairs. For Chinese-to-English NIST task, there are about 1% of the rules that do not have contexts; for German-to-English task, this number is about 0.4%. We assign a uniform number as their bilingual sense similarity score, and this number is tuned through MERT. We call it the *null context* feature. It is included in all the results reported from Table 2 to Table 6. In Table 7, we show the weight of the *null context* feature tuned by running MERT in the experiments reported in Section 5.2. We can learn that penalties always discourage using those rules which have no context to be extracted.

	Task		
Alg.	CE_SD	CE_LD	DE
Alg2 IBM	-0.09	-0.37	-0.15
Alg2 COS	-0.59	-0.42	-0.36

Table 7: Weight learned for employing the *null context* feature. CE_SD, CE_LD and DE are Chinese-to-English *small data* task, *large data* task and German-to-English task respectively.

6.5 Discussion

Our aim in this paper is to characterize the semantic similarity of bilingual hierarchical rules. We can make several observations concerning our features:

1) Rules that are largely syntactic in nature, such as *的 X ||| the X of*, will have very diffuse "meanings" and therefore lower similarity scores. It could be that the gains we obtained come simply from biasing the system against such rules. However, the results in table 6 show that this is unlikely to be the case: features that just count context words help very little.

2) In addition to bilingual similarity, Alg2 relies on the degree of monolingual similarity between the sense of a source or target unit within a rule, and the sense of the unit in general. This has a bias in favor of less ambiguous rules, i.e. rules involving only units with closely related meanings. Although this bias is helpful on its own, possibly due to the mechanism we outline in section 6.1, it appears to have a synergistic effect when used along with the bilingual similarity feature.

3) Finally, we note that many of the features we use for capturing similarity, such as the context "*the, of*" for instantiations of *X* in the unit *the X of*, are arguably more syntactic than semantic. Thus, like other "semantic" approaches, ours can be seen as blending syntactic and semantic information.

7 Related Work

There has been extensive work on incorporating semantics into SMT. Key papers by Carpuat and Wu (2007) and Chan et al (2007) showed that word-sense disambiguation (WSD) techniques relying on source-language context can be effective in selecting translations in phrase-based and hierarchical SMT. More recent work has aimed at incorporating richer disambiguating features into the SMT log-linear model (Gimpel and Smith, 2008; Chiang et al, 2009); predicting coherent sets of target words rather than individual phrase translations (Bangalore et al, 2009; Mauser et al, 2009); and selecting applicable rules in hierarchical (He et al, 2008) and syntactic (Liu et al, 2008) translation, relying on source as well as target context. Work by Wu and Fung (2009) breaks new ground in attempting to match semantic roles derived from a semantic parser across source and target languages.

Our work is different from all the above approaches in that we attempt to discriminate among hierarchical rules based on: 1) the degree of bilingual semantic similarity between source and target translation units; and 2) the monolingual semantic similarity between occurrences of source or target units as part of the given rule, and in general. In another words, WSD explicitly tries to choose a translation given the current source context, while our work rates rule pairs independent of the current context.

8 Conclusions and Future Work

In this paper, we have proposed an approach that uses the vector space model to compute the sense

similarity for terms from parallel corpora and applied it to statistical machine translation. We saw that the bilingual sense similarity computed by our algorithm led to significant improvements. Therefore, we can answer the questions proposed in Section 1. We have shown that the sense similarity computed between units from parallel corpora by means of our algorithm is helpful for at least one multilingual application: statistical machine translation.

Finally, although we described and evaluated bilingual sense similarity algorithms applied to a hierarchical phrase-based system, this method is also suitable for syntax-based MT systems and phrase-based MT systems. The only difference is the definition of the context. For a syntax-based system, the context of a rule could be defined similarly to the way it was defined in the work described above. For a phrase-based system, the context of a phrase could be defined as its surrounding words in a given size window. In our future work, we may try this algorithm on syntax-based MT systems and phrase-based MT systems with different context features. It would also be possible to use this technique during training of an SMT system – for instance, to improve the bilingual word alignment or reduce the training data noise.

References

S. Bangalore, S. Kanthak, and P. Haffner. 2009. Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction. In: Goutte et al (ed.), *Learning Machine Translation*. MIT Press.

P. F. Brown, V. J. Della Pietra, S. A. Della Pietra & R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2) 263-312.

J. Bullinaria and J. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39 (3), 510–526.

M. Carpuat and D. Wu. 2007. Improving Statistical Machine Translation using Word Sense Disambiguation. In: *Proceedings of EMNLP*, Prague.

M. Carpuat. 2009. One Translation per Discourse. In: *Proceedings of NAACL HLT Workshop on Semantic Evaluations*, Boulder, CO.

Y. Chan, H. Ng and D. Chiang. 2007. Word Sense Disambiguation Improves Statistical Machine Translation. In: *Proceedings of ACL*, Prague.

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In: *Proceedings of ACL*, pp. 263–270.

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

D. Chiang, W. Wang and K. Knight. 2009. 11,001 new features for statistical machine translation. In: *Proc. NAACL HLT*, pp. 218–226.

K. W. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

W. B. Frakes and R. Baeza-Yates, editors. 1992. *Information Retrieval, Data Structure and Algorithms*. Prentice Hall.

P. Fung. 1998. A statistical view on bilingual lexicon extraction: From parallel corpora to non-parallel corpora. In: *Proceedings of AMTA*, pp. 1–17. Oct. Langhorne, PA, USA.

J. Gimenez and L. Marquez. 2009. Discriminative Phrase Selection for SMT. In: Goutte et al (ed.), *Learning Machine Translation*. MIT Press.

K. Gimpel and N. A. Smith. 2008. Rich Source-Side Context for Statistical Machine Translation. In: *Proceedings of WMT*, Columbus, OH.

Z. Harris. 1954. Distributional structure. *Word*, 10(23): 146-162.

Z. He, Q. Liu, and S. Lin. 2008. Improving Statistical Machine Translation using Lexicalized Rule Selection. In: *Proceedings of COLING*, Manchester, UK.

D. Hindle. 1990. Noun classification from predicate-argument structures. In: *Proceedings of ACL*. pp. 268-275. Pittsburgh, PA.

P. Koehn, F. Och, D. Marcu. 2003. Statistical Phrase-Based Translation. In: *Proceedings of HLT-NAACL*. pp. 127-133, Edmonton, Canada

P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In: *Proceedings of EMNLP*, pp. 388–395. July, Barcelona, Spain.

T. Landauer and S. T. Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*. 104:211-240.

Z. Li, C. Callison-Burch, C. Dyer, J. Ganitkevitch, S. Khudanpur, L. Schwartz, W. Thornton, J. Weese and O. Zaidan, 2009. Joshua: An Open Source Toolkit for Parsing-based Machine Translation. In: *Proceedings of the WMT*. March. Athens, Greece.

D. Lin. 1998. Automatic retrieval and clustering of similar words. In: *Proceedings of COLING/ACL-98*. pp. 768-774. Montreal, Canada.

Q. Liu, Z. He, Y. Liu and S. Lin. 2008. Maximum Entropy based Rule Selection Model for Syntax-based Statistical Machine Translation. In: *Proceedings of EMNLP*, Honolulu, Hawaii.

K. Lund, and C. Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*, 28 (2), 203–208.

A. Mauser, S. Hasan and H. Ney. 2009. Extending Statistical Machine Translation with Discriminative and Trigger-Based Lexicon Models. In: *Proceedings of EMNLP*, Singapore.

F. Och. 2003. Minimum error rate training in statistical machine translation. In: *Proceedings of ACL*. Sapporo, Japan.

S. Pado and M. Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33 (2), 161–199.

P. Pantel and D. Lin. 2002. Discovering word senses from text. In: *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 613–619. Edmonton, Canada.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pp. 311–318. July. Philadelphia, PA, USA.

R. Rapp. 1999. Automatic Identification of Word Translations from Unrelated English and German Corpora. In: *Proceedings of ACL*, pp. 519–526. June. Maryland.

G. Salton and M. J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.

P. Turney. 2001. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In: *Proceedings of the Twelfth European Conference on Machine Learning*, pp. 491–502, Berlin, Germany.

D. Wu and P. Fung. 2009. Semantic Roles for SMT: A Hybrid Two-Pass Model. In: *Proceedings of NAACL/HLT*, Boulder, CO.

D. Yuret and M. A. Yatbaz. 2009. The Noisy Channel Model for Unsupervised Word Sense Disambiguation. In: *Computational Linguistics*. Vol. 1(1) 1-18.

R. Zens and H. Ney. 2004. Improvements in phrase-based statistical machine translation. In: *Proceedings of NAACL-HLT*. Boston, MA.

B. Zhao, S. Vogel, M. Eck, and A. Waibel. 2004. Phrase pair rescoring with term weighting for statistical machine translation. In *Proceedings of EMNLP*, pp. 206–213. July. Barcelona, Spain.