Word Sense Disambiguation Improves Statistical Machine Translation

Yee Seng Chan and **Hwee Tou Ng**
Department of Computer Science
National University of Singapore
3 Science Drive 2
Singapore 117543
{chanys, nght}@comp.nus.edu.sg

David Chiang
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292, USA
chiang@isi.edu

Abstract

Recent research presents conflicting evidence on whether word sense disambiguation (WSD) systems can help to improve the performance of statistical machine translation (MT) systems. In this paper, we successfully integrate a state-of-the-art WSD system into a state-of-the-art hierarchical phrase-based MT system, Hiero. We show for the first time that integrating a WSD system improves the performance of a state-of-the-art statistical MT system on an actual translation task. Furthermore, the improvement is statistically significant.

1 Introduction

Many words have multiple meanings, depending on the context in which they are used. Word sense disambiguation (WSD) is the task of determining the correct meaning or sense of a word in context. WSD is regarded as an important research problem and is assumed to be helpful for applications such as machine translation (MT) and information retrieval.

In translation, different senses of a word *w* in a source language may have different translations in a target language, depending on the particular meaning of *w* in context. Hence, the assumption is that in resolving sense ambiguity, a WSD system will be able to help an MT system to determine the correct translation for an ambiguous word. To determine the correct sense of a word, WSD systems typically use a wide array of features that are not limited to the local context of *w*, and some of these features may not be used by state-of-the-art statistical MT systems.

To perform translation, state-of-the-art MT systems use a statistical phrase-based approach (Marcu and Wong, 2002; Koehn et al., 2003; Och and Ney, 2004) by treating phrases as the basic units of translation. In this approach, a phrase can be any sequence of consecutive words and is not necessarily linguistically meaningful. Capitalizing on the strength of the phrase-based approach, Chiang (2005) introduced a *hierarchical* phrase-based statistical MT system, Hiero, which achieves significantly better translation performance than Pharaoh (Koehn, 2004a), which is a state-of-the-art phrase-based statistical MT system.

Recently, some researchers investigated whether performing WSD will help to improve the performance of an MT system. Carpuat and Wu (2005) integrated the translation predictions from a Chinese WSD system (Carpuat et al., 2004) into a Chinese-English word-based statistical MT system using the ISI ReWrite decoder (Germann, 2003). Though they acknowledged that directly using English translations as word senses would be ideal, they instead predicted the HowNet sense of a word and then used the English gloss of the HowNet sense as the WSD model's predicted translation. They did not incorporate their WSD model or its predictions into their translation model; rather, they used the WSD predictions either to constrain the options available to their decoder, or to postedit the output of their decoder. They reported the negative result that WSD decreased the performance of MT based on their experiments.

In another work (Vickrey et al., 2005), the WSD problem was recast as a *word translation* task. The

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 33–40,
Prague, Czech Republic, June 2007.

translation choices for a word *w* were defined as the set of words or phrases aligned to *w*, as gathered from a word-aligned parallel corpus. The authors showed that they were able to improve their model's accuracy on two simplified translation tasks: word translation and blank-filling.

Recently, Cabezas and Resnik (2005) experimented with incorporating WSD translations into Pharaoh, a state-of-the-art phrase-based MT system (Koehn et al., 2003). Their WSD system provided additional translations to the phrase table of Pharaoh, which fired a new model feature, so that the decoder could weigh the additional alternative translations against its own. However, they could not automatically tune the weight of this feature in the same way as the others. They obtained a relatively small improvement, and no statistical significance test was reported to determine if the improvement was statistically significant.

Note that the experiments in (Carpuat and Wu, 2005) did not use a state-of-the-art MT system, while the experiments in (Vickrey et al., 2005) were not done using a full-fledged MT system and the evaluation was not on how well each source sentence was translated as a whole. The relatively small improvement reported by Cabezas and Resnik (2005) without a statistical significance test appears to be inconclusive. Considering the conflicting results reported by prior work, it is not clear whether a WSD system can help to improve the performance of a state-of-the-art statistical MT system.

In this paper, we successfully integrate a state-of-the-art WSD system into the state-of-the-art hierarchical phrase-based MT system, Hiero (Chiang, 2005). The integration is accomplished by introducing two additional features into the MT model which operate on the existing rules of the grammar, without introducing competing rules. These features are treated, both in feature-weight tuning and in decoding, on the same footing as the rest of the model, allowing it to weigh the WSD model predictions against other pieces of evidence so as to optimize translation accuracy (as measured by BLEU). The contribution of our work lies in showing for the first time that integrating a WSD system significantly improves the performance of a state-of-the-art statistical MT system on an actual translation task.

In the next section, we describe our WSD system. Then, in Section 3, we describe the Hiero MT system and introduce the two new features used to integrate the WSD system into Hiero. In Section 4, we describe the training data used by the WSD system. In Section 5, we describe how the WSD translations provided are used by the decoder of the MT system. In Section 6 and 7, we present and analyze our experimental results, before concluding in Section 8.

2 Word Sense Disambiguation

Prior research has shown that using Support Vector Machines (SVM) as the learning algorithm for WSD achieves good results (Lee and Ng, 2002). For our experiments, we use the SVM implementation of (Chang and Lin, 2001) as it is able to work on multi-class problems to output the classification probability for each class.

Our implemented WSD classifier uses the knowledge sources of local collocations, parts-of-speech (POS), and surrounding words, following the successful approach of (Lee and Ng, 2002). For local collocations, we use 3 features, $w_{-1}w_{+1}$, w_{-1}, and w_{+1}, where w_{-1} (w_{+1}) is the token immediately to the left (right) of the current ambiguous word occurrence w. For parts-of-speech, we use 3 features, P_{-1}, P_0, and P_{+1}, where P_0 is the POS of w, and P_{-1} (P_{+1}) is the POS of w_{-1} (w_{+1}). For surrounding words, we consider all unigrams (single words) in the surrounding context of *w*. These unigrams can be in a different sentence from *w*. We perform feature selection on surrounding words by including a unigram only if it occurs 3 or more times in some sense of *w* in the training data.

To measure the accuracy of our WSD classifier, we evaluate it on the test data of SENSEVAL-3 Chinese lexical-sample task. We obtain accuracy that compares favorably to the best participating system in the task (Carpuat et al., 2004).

3 Hiero

Hiero (Chiang, 2005) is a hierarchical phrase-based model for statistical machine translation, based on weighted synchronous context-free grammar (CFG) (Lewis and Stearns, 1968). A synchronous CFG consists of rewrite rules such as the following:

$$\mathrm{X} \rightarrow \langle \gamma, \alpha \rangle \qquad (1)$$

where X is a non-terminal symbol, γ (α) is a string of terminal and non-terminal symbols in the source (target) language, and there is a one-to-one correspondence between the non-terminals in γ and α indicated by co-indexation. Hence, γ and α always have the same number of non-terminal symbols. For instance, we could have the following grammar rule:

$$\text{X} \rightarrow \langle \text{每 月 到 X}_{\boxed{1}}, \text{go to X}_{\boxed{1}} \text{ every month to} \rangle \quad (2)$$

where boxed indices represent the correspondences between non-terminal symbols.

Hiero extracts the synchronous CFG rules automatically from a word-aligned parallel corpus. To translate a source sentence, the goal is to find its most probable derivation using the extracted grammar rules. Hiero uses a general log-linear model (Och and Ney, 2002) where the weight of a derivation D for a particular source sentence and its translation is

$$w(D) = \prod_i \phi_i(D)^{\lambda_i} \quad (3)$$

where ϕ_i is a feature function and λ_i is the weight for feature ϕ_i. To ensure efficient decoding, the ϕ_i are subject to certain locality restrictions. Essentially, they should be defined as products of functions defined on isolated synchronous CGF rules; however, it is possible to extend the domain of locality of the features somewhat. A *n*-gram language model adds a dependence on (*n*$-$1) neighboring target-side words (Wu, 1996; Chiang, 2007), making decoding much more difficult but still polynomial; in this paper, we add features that depend on the neighboring *source-side* words, which does not affect decoding complexity at all because the source string is fixed. In principle we could add features that depend on arbitrary source-side context.

3.1 New Features in Hiero for WSD

To incorporate WSD into Hiero, we use the translations proposed by the WSD system to help Hiero obtain a better or more probable derivation during the translation of each source sentence. To achieve this, when a grammar rule *R* is considered during decoding, and we recognize that some of the terminal symbols (words) in α are also chosen by the WSD system as translations for some terminal symbols (words) in γ, we compute the following features:

- $P_{wsd}(t \mid s)$ gives the contextual probability of the WSD classifier choosing t as a translation for s, where t (s) is some substring of terminal symbols in α (γ). Because this probability only applies to some rules, and we don't want to penalize those rules, we must add another feature,

- $Pty_{wsd} = \exp(-|t|)$, where t is the translation chosen by the WSD system. This feature, with a negative weight, rewards rules that use translations suggested by the WSD module.

Note that we can take the negative logarithm of the rule/derivation weights and think of them as costs rather than probabilities.

4 Gathering Training Examples for WSD

Our experiments were for Chinese to English translation. Hence, in the context of our work, a synchronous CFG grammar rule X $\rightarrow \langle\gamma, \alpha\rangle$ gathered by Hiero consists of a Chinese portion γ and a corresponding English portion α, where each portion is a sequence of words and non-terminal symbols.

Our WSD classifier suggests a list of English phrases (where each phrase consists of one or more English words) with associated contextual probabilities as possible translations for each particular Chinese phrase. In general, the Chinese phrase may consist of *k* Chinese words, where $k = 1, 2, 3, \ldots$. However, we limit k to 1 or 2 for experiments reported in this paper. Future work can explore enlarging *k*.

Whenever Hiero is about to extract a grammar rule where its Chinese portion is a phrase of one or two Chinese words with no non-terminal symbols, we note the location (sentence and token offset) in the Chinese half of the parallel corpus from which the Chinese portion of the rule is extracted. The actual sentence in the corpus containing the Chinese phrase, and the one sentence before and the one sentence after that actual sentence, will serve as the context for one training example for the Chinese phrase, with the corresponding English phrase of the grammar rule as its translation. Hence, unlike traditional WSD where the sense classes are tied to a specific sense inventory, our "senses" here consist of the English phrases extracted as translations for each Chinese phrase. Since the extracted training data may be noisy, for each Chinese phrase, we remove English translations that occur only once. Furthermore, we only attempt WSD classification for those Chinese phrases with at least 10 training examples.

Using the WSD classifier described in Section 2, we classified the words in each Chinese source sentence to be translated. We first performed WSD on all single Chinese words which are either noun, verb, or adjective. Next, we classified the Chinese phrases consisting of 2 consecutive Chinese words by simply treating the phrase as *a single unit*. When performing classification, we give as output the set of English translations with associated context-dependent probabilities, which are the probabilities of a Chinese word (phrase) translating into each English phrase, depending on the context of the Chinese word (phrase). After WSD, the *i*th word c_i in every Chinese sentence may have up to 3 sets of associated translations provided by the WSD system: a set of translations for c_i as a single word, a second set of translations for $c_{i-1}c_i$ considered as a single unit, and a third set of translations for c_ic_{i+1} considered as a single unit.

5 Incorporating WSD during Decoding

The following tasks are done for each rule that is considered during decoding:

- identify Chinese words to suggest translations for
- match suggested translations against the English side of the rule
- compute features for the rule

The WSD system is able to predict translations only for a subset of Chinese words or phrases. Hence, we must first identify which parts of the Chinese side of the rule have suggested translations available. Here, we consider substrings of length up to two, and we give priority to longer substrings.

Next, we want to know, for each Chinese substring considered, whether the WSD system supports the Chinese-English translation represented by the rule. If the rule is finally chosen as part of the best derivation for translating the Chinese sentence, then all the words in the English side of the rule will appear in the translated English sentence. Hence, we need to match the translations suggested by the WSD system against the English side of the rule. It is for these matching rules that the WSD features will apply.

The translations proposed by the WSD system may be more than one word long. In order for a proposed translation to match the rule, we require two conditions. First, the proposed translation must be a substring of the English side of the rule. For example, the proposed translation "every to" would not match the chunk "every month to". Second, the match must contain at least one aligned Chinese-English word pair, but we do not make any other requirements about the alignment of the other Chinese or English words.[1] If there are multiple possible matches, we choose the longest proposed translation; in the case of a tie, we choose the proposed translation with the highest score according to the WSD model.

Define a *chunk* of a rule to be a maximal substring of terminal symbols on the English side of the rule. For example, in Rule (2), the chunks would be "go to" and "every month to". Whenever we find a matching WSD translation, we mark the whole chunk on the English side as consumed.

Finally, we compute the feature values for the rule. The feature $P_{wsd}(t \mid s)$ is the sum of the costs (according to the WSD model) of all the matched translations, and the feature Pty_{wsd} is the sum of the lengths of all the matched translations.

Figure 1 shows the pseudocode for the rule scoring algorithm in more detail, particularly with regards to resolving conflicts between overlapping matches. To illustrate the algorithm given in Figure 1, consider Rule (2). Hereafter, we will use symbols to represent the Chinese and English words in the rule: c_1, c_2, and c_3 will represent the Chinese words "每", "月", and "到" respectively. Similarly, e_1, e_2, e_3, e_4, and e_5 will represent the English words *go*, *to*, *every*, *month*, and *to* respectively. Hence, Rule (2) has two chunks: e_1e_2 and $e_3e_4e_5$. When the rule is extracted from the parallel corpus, it has these alignments between the words of its Chinese and English portion: $\{c_1\text{–}e_3, c_2\text{–}e_4, c_3\text{–}e_1, c_3\text{–}e_2, c_3\text{–}e_5\}$, which means that c_1 is aligned to e_3, c_2 is aligned to

[1] In order to check this requirement, we extended Hiero to make word alignment information available to the decoder.

```
Input: rule R considered during decoding with its own associated $cost_R$
$L_c$ = list of symbols in Chinese portion of R
WSDcost = 0
i = 1
while i $\leq$ len($L_c$):
    $c_i$ = ith symbol in $L_c$
    if $c_i$ is a Chinese word (i.e., not a non-terminal symbol):
        seenChunk = $\emptyset$    // seenChunk is a global variable and is passed by reference to matchWSD
        if ($c_i$ is not the last symbol in $L_c$) and ($c_{i+1}$ is a terminal symbol): then $c_{i+1}$=(i+1)th symbol in $L_c$, else $c_{i+1}$ = NULL
        if ($c_{i+1}$!=NULL) and ($c_i$, $c_{i+1}$) as a single unit has WSD translations:
            $WSD_c$ = set of WSD translations for ($c_i$, $c_{i+1}$) as a single unit with context-dependent probabilities
            WSDcost = WSDcost + matchWSD($c_i$, $WSD_c$, seenChunk)
            WSDcost = WSDcost + matchWSD($c_{i+1}$, $WSD_c$, seenChunk)
            i = i + 1
        else:
            $WSD_c$ = set of WSD translations for $c_i$ with context-dependent probabilities
            WSDcost = WSDcost + matchWSD($c_i$, $WSD_c$, seenChunk)
    i = i + 1
$cost_R$ = $cost_R$ + WSDcost

matchWSD(c, $WSD_c$, seenChunk):
    // seenChunk is the set of chunks of R already examined for possible matching WSD translations
    cost = 0
    ChunkSet = set of chunks in R aligned to c
    for $chunk_j$ in ChunkSet:
        if $chunk_j$ not in seenChunk:
            seenChunk = seenChunk $\cup$ { $chunk_j$ }
            $E_{chunk_j}$ = set of English words in $chunk_j$ aligned to c
            $Candidate_{wsd}$ = $\emptyset$
            for $wsd_k$ in $WSD_c$:
                if ($wsd_k$ is sub-sequence of $chunk_j$) and ($wsd_k$ contains at least one word in $E_{chunk_j}$)
                    $Candidate_{wsd}$ = $Candidate_{wsd}$ $\cup$ { $wsd_k$ }
            $wsd_{best}$ = best matching translation in $Candidate_{wsd}$ against $chunk_j$
            cost = cost + costByWSDfeatures($wsd_{best}$)    // costByWSDfeatures sums up the cost of the two WSD features
    return cost
```

Figure 1: WSD translations affecting the cost of a rule *R* considered during decoding.

e_4, and c_3 is aligned to e_1, e_2, and e_5. Although all words are aligned here, in general for a rule, some of its Chinese or English words may not be associated with any alignments.

In our experiment, c_1c_2 as a phrase has a list of translations proposed by the WSD system, including the English phrase "every month". *matchWSD* will first be invoked for c_1, which is aligned to only one chunk $e_3e_4e_5$ via its alignment with e_3. Since "every month" is a sub-sequence of the chunk and also contains the word e_3 ("every"), it is noted as a candidate translation. Later, it is determined that the most number of words any candidate translation has is two words. Since among all the 2-word candidate translations, the translation "every month" has the highest translation probability as assigned by the WSD classifier, it is chosen as the best matching translation for the chunk. *matchWSD* is then invoked for c_2, which is aligned to only one chunk $e_3e_4e_5$. However, since this chunk has already been examined by c_1 with which it is considered as a phrase, no further matching is done for c_2. Next, *matchWSD* is invoked for c_3, which is aligned to both chunks of *R*. The English phrases "go to" and "to" are among the list of translations proposed by the WSD system for c_3, and they are eventually chosen as the best matching translations for the chunks e_1e_2 and $e_3e_4e_5$, respectively.

6 Experiments

As mentioned, our experiments were on Chinese to English translation. Similar to (Chiang, 2005), we trained the Hiero system on the FBIS corpus, used the NIST MT 2002 evaluation test set as our development set to tune the feature weights, and the NIST MT 2003 evaluation test set as our test data. Using

System	BLEU-4	Individual *n*-gram precisions			
		1	2	3	4
Hiero	29.73	74.73	40.14	21.83	11.93
Hiero+WSD	30.30	74.82	40.40	22.45	12.42

Table 1: BLEU scores

System	Features									
	$P_{lm}(e)$	$P(\gamma\|\alpha)$	$P(\alpha\|\gamma)$	$P_w(\gamma\|\alpha)$	$P_w(\alpha\|\gamma)$	Pty_{phr}	$Glue$	Pty_{word}	$P_{wsd}(t\|s)$	Pty_{wsd}
Hiero	0.2337	0.0882	0.1666	0.0393	0.1357	0.0665	-0.0582	-0.4806	-	-
Hiero+WSD	0.1937	0.0770	0.1124	0.0487	0.0380	0.0988	-0.0305	-0.1747	0.1051	-0.1611

Table 2: Weights for each feature obtained by MERT training. The first eight features are those used by Hiero in (Chiang, 2005).

the English portion of the FBIS corpus and the Xinhua portion of the Gigaword corpus, we trained a trigram language model using the SRI Language Modelling Toolkit (Stolcke, 2002). Following (Chiang, 2005), we used the version 11a NIST BLEU script with its default settings to calculate the BLEU scores (Papineni et al., 2002) based on case-insensitive *n*-gram matching, where *n* is up to 4.

First, we performed word alignment on the FBIS parallel corpus using GIZA++ (Och and Ney, 2000) in both directions. The word alignments of both directions are then combined into a single set of alignments using the “diag-and” method of Koehn et al. (2003). Based on these alignments, synchronous CFG rules are then extracted from the corpus. While Hiero is extracting grammar rules, we gathered WSD training data by following the procedure described in section 4.

6.1 Hiero Results

Using the MT 2002 test set, we ran the minimum-error rate training (MERT) (Och, 2003) with the decoder to tune the weights for each feature. The weights obtained are shown in the row *Hiero* of Table 2. Using these weights, we run Hiero's decoder to perform the actual translation of the MT 2003 test sentences and obtained a BLEU score of 29.73, as shown in the row *Hiero* of Table 1. This is higher than the score of 28.77 reported in (Chiang, 2005), perhaps due to differences in word segmentation, etc. Note that comparing with the MT systems used in (Carpuat and Wu, 2005) and (Cabezas and Resnik, 2005), the Hiero system we are using represents a much stronger baseline MT system upon which the WSD system must improve.

6.2 Hiero+WSD Results

We then added the WSD features of Section 3.1 into Hiero and reran the experiment. The weights obtained by MERT are shown in the row *Hiero+WSD* of Table 2. We note that a negative weight is learnt for Pty_{wsd}. This means that in general, the model prefers grammar rules having chunks that matches WSD translations. This matches our intuition. Using the weights obtained, we translated the test sentences and obtained a BLEU score of **30.30**, as shown in the row *Hiero+WSD* of Table 1. The improvement of 0.57 is statistically significant at $p < 0.05$ using the sign-test as described by Collins et al. (2005), with 374 (+1), 318 (-1) and 227 (0). Using the bootstrap-sampling test described in (Koehn, 2004b), the improvement is statistically significant at $p < 0.05$. Though the improvement is modest, it is statistically significant and this positive result is important in view of the negative findings in (Carpuat and Wu, 2005) that WSD does not help MT. Furthermore, note that Hiero+WSD has higher *n*-gram precisions than Hiero.

7 Analysis

Ideally, the WSD system should be suggesting high-quality translations which are frequently part of the reference sentences. To determine this, we note the set of grammar rules used in the best derivation for translating each test sentence. From the rules of each test sentence, we tabulated the set of translations proposed by the WSD system and check whether they are found in the associated reference sentences.

On the entire set of NIST MT 2003 evaluation test sentences, an average of 10.36 translations proposed

No. of words in WSD translations	All test sentences		+1 from Collins sign-test	
	No. of WSD translations used	% match reference	No. of WSD translations used	% match reference
1	7087	77.31	3078	77.68
2	1930	66.11	861	64.92
3	371	43.13	171	48.54
4	124	26.61	52	28.85

Table 3: Number of WSD translations used and proportion that matches against respective reference sentences. WSD translations longer than 4 words are very sparse (less than 10 occurrences) and thus they are not shown.

by the WSD system were used for each sentence. When limited to the set of 374 sentences which were judged by the Collins sign-test to have better translations from Hiero+WSD than from Hiero, a higher number (11.14) of proposed translations were used on average. Further, for the entire set of test sentences, 73.01% of the proposed translations are found in the reference sentences. This increased to a proportion of 73.22% when limited to the set of 374 sentences. These figures show that having more, and higher-quality proposed translations contributed to the set of 374 sentences being better translations than their respective original translations from Hiero. Table 3 gives a detailed breakdown of these figures according to the number of words in each proposed translation. For instance, over all the test sentences, the WSD module gave 7087 translations of single-word length, and 77.31% of these translations match their respective reference sentences. We note that although the proportion of matching 2-word translations is slightly lower for the set of 374 sentences, the proportion increases for translations having more words.

After the experiments in Section 6 were completed, we visually inspected the translation output of Hiero and Hiero+WSD to categorize the ways in which integrating WSD contributes to better translations. The first way in which WSD helps is when it enables the integrated Hiero+WSD system to output extra appropriate English words. For example, the translations for the Chinese sentence "…或 其 他 「恶劣 行为 」， 将 无法 取得 更 多 援助 或 其他 让步 。" are as follows.

- Hiero: *… or other bad behavior ", will be more aid and other concessions.*

- Hiero+WSD: *… or other bad behavior ", will be unable to obtain more aid and other concessions.*

Here, the Chinese words "无法 取得" are not translated by Hiero at all. By providing the correct translation of "*unable to obtain*" for "无法 取得", the translation output of Hiero+WSD is more complete.

A second way in which WSD helps is by correcting a previously incorrect translation. For example, for the Chinese sentence "…， 在 全 国 各族 人 民 ， … ", the WSD system helps to correct Hiero's original translation by providing the correct translation of "*all ethnic groups*" for the Chinese phrase "各 族":

- Hiero: *…, and people of all nationalities across the country, …*

- Hiero+WSD: *…, and people of all ethnic groups across the country, …*

We also looked at the set of 318 sentences that were judged by the Collins sign-test to be worse translations. We found that in some situations, Hiero+WSD has provided extra appropriate English words, but those particular words are not used in the reference sentences. An interesting example is the translation of the Chinese sentence "澳洲 外长 指 北韩 行为 恶劣 将 无法 取得 更 多 援助".

- Hiero: *Australian foreign minister said that North Korea bad behavior will be more aid*

- Hiero+WSD: *Australian foreign minister said that North Korea bad behavior will be unable to obtain more aid*

This is similar to the example mentioned earlier. In this case however, those extra English words provided by Hiero+WSD, though appropriate, do not result in more *n*-gram matches as the reference sentences used phrases such as "*will not gain*", "*will not get*", etc. Since the BLEU metric is precision based, the longer sentence translation by Hiero+WSD gets a lower BLEU score instead.

8 Conclusion

We have shown that WSD improves the translation performance of a state-of-the-art hierarchical phrase-based statistical MT system and this improvement is statistically significant. We have also demonstrated one way to integrate a WSD system into an MT system without introducing any rules that compete against existing rules, and where the feature-weight tuning and decoding place the WSD system on an equal footing with the other model components. For future work, an immediate step would be for the WSD classifier to provide translations for longer Chinese phrases. Also, different alternatives could be tried to match the translations provided by the WSD classifier against the chunks of rules. Finally, besides our proposed approach of integrating WSD into statistical MT via the introduction of two new features, we could explore other alternative ways of integration.

Acknowledgements

Yee Seng Chan is supported by a Singapore Millennium Foundation Scholarship (ref no. SMF-2004-1076). David Chiang was partially supported under the GALE program of the Defense Advanced Research Projects Agency, contract HR0011-06-C-0022.

References

C. Cabezas and P. Resnik. 2005. Using WSD techniques for lexical selection in statistical machine translation. Technical report, University of Maryland.

M. Carpuat and D. Wu. 2005. Word sense disambiguation vs. statistical machine translation. In *Proc. of ACL05*, pages 387–394.

M. Carpuat, W. Su, and D. Wu. 2004. Augmenting ensemble classification for word sense disambiguation with a kernel PCA model. In *Proc. of SENSEVAL-3*, pages 88–92.

C. C. Chang and C. J. Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/˜cjlin/libsvm`.

D. Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL05*, pages 263–270.

D. Chiang. 2007. Hierarchical phrase-based translation. *To appear in Computational Linguistics*, 33(2).

M. Collins, P. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proc. of ACL05*, pages 531–540.

U. Germann. 2003. Greedy decoding for statistical machine translation in almost linear time. In *Proc. of HLT-NAACL03*, pages 72–79.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of HLT-NAACL03*, pages 48–54.

P. Koehn. 2003. *Noun Phrase Translation*. Ph.D. thesis, University of Southern California.

P. Koehn. 2004a. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proc. of AMTA04*, pages 115–124.

P. Koehn. 2004b. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP04*, pages 388–395.

Y. K. Lee and H. T. Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proc. of EMNLP02*, pages 41–48.

P. M. II Lewis and R. E. Stearns. 1968. Syntax-directed transduction. *Journal of the ACM*, 15(3):465–488.

D. Marcu and W. Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proc. of EMNLP02*, pages 133–139.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proc. of ACL00*, pages 440–447.

F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of ACL02*, pages 295–302.

F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL03*, pages 160–167.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proc. of ACL02*, pages 311–318.

A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. of ICSLP02*, pages 901–904.

D. Vickrey, L. Biewald, M. Teyssier, and D. Koller. 2005. Word-sense disambiguation for machine translation. In *Proc. of EMNLP05*, pages 771–778.

D. Wu. 1996. A polynomial-time algorithm for statistical machine translation. In *Proc. of ACL96*, pages 152–158.