Word-Sense Disambiguation for Machine Translation

Abstract

In word sense disambiguation, a system attempts to determine the sense of word from contextual features. Major barriers to building a high-performing word sense disambiguation system include the difficulty of labeling data for this task and of predicting fine-grained sense distinctions. In contrast, we can use parallel language corpora as a large supply of potential data. In this paper we present algorithms for solving the word translation problem and demonstrate a significant improvement over a baseline system. The predictions resulting from this system can then be used to inform a standard machine translation system.

1 Introduction

The problem of distinguishing between multiple possible senses of a word is an important subtask in many NLP applications. However, despite its conceptual simplicity, and its obvious formulation as a standard classification problem, achieving high levels of performance on this task has been a remarkably elusive goal.

In its standard formulation, the disambiguation task is specified via an ontology defining the different senses of ambiguous words. In the Senseval competition, for example, WordNet (Fellbaum, 1998) is used to define this ontology. However, ontologies such as WordNet are not ideally suited to the task of word-sense disambiguation. In many cases, WordNet is overly "specific", defining senses which are very similar and hard to distinguish. For example, there are seven definitions of "respect" *as a noun* (Table 1; there are even more when the verb definitions are included as well. Such closely related senses pose a challenge both for automatic disambiguation and hand labeling. Moreover, the use of a very fine-grained set of senses, most of which are quite rare in practice, makes it very difficult to obtain sufficient amounts of training data.

These issues are clearly reflected in the performance of current word-sense disambiguation systems. When given a large amount of training data for a particular word with reasonably clear sense distinctions, existing systems perform fairly well. However, for the "all-words" task, where all ambiguous words from a test corpus must be disambiguated, it has so far proven difficult to perform significantly better than the baseline heuristic of choosing the most common sense for each word[1].

In this paper, we propose a different formulation of the word-sense disambiguation task. Rather than considering this task on its own, we consider a task of disambiguating words for the purpose of some larger goal. Clearly, word-sense disambiguation is important for many natural language tasks, but perhaps the most direct and compelling application of a working word-sense disambiguator is to machine translation. If we knew the correct semantic meaning of each word in the source language, we could more accurately determine the appropriate words in the target language. Importantly, for this application, subtle shades of meaning will often be irrelevant in choosing the most appropriate words in the target language, as closely related senses of a single word in one language are often encoded by a single word in another. Thus, in the context of this larger goal, we can focus only on sense distinctions that a human would consider when choosing the translation of a

[1] See results of Senseval-2, available at http://www.sle.sharp.co.uk/senseval2

WordNet Synset	Definition
respect, regard	(usually preceded by 'in') a detail or point
esteem, regard, respect	the condition of being honored (esteemed or respected or well regarded)
respect, esteem, regard	an attitude of admiration or esteem
deference, respect	a courteous expression (by word or deed) of esteem or regard
obedience, respect	behavior intended to please your parents
regard, respect	a feeling of friendship and esteem
deference, respect, respectfulness	courteous regard for people's feelings

Table 1: WordNet Noun Synsets containing "respect"

word in the source language.

We therefore consider the task of word-sense disambiguation for the purpose of machine translation. Instead of trying to predict the sense of a particular word w, we predict the possible translations of w into the target language. We both train and evaluate the system on this task. This formulation of the word-sense disambiguation task, which we will to refer to as *word translation*, has multiple advantages. First, a very large amount of "partially-labeled" data is available for this task in the form of bilingual corpora (which exist for a wide range of languages). Second, the "labeling" of these corpora (that is, translation from one language to another), is a task at which humans are quite proficient and which does not generally require the labeler (translator) to make difficult distinctions between fine shades of meaning.

In the remainder of this paper, we first define how training data for this task can be acquired automatically from bilingual corpora. We then provide learning algorithms for word translation. Although our algorithms can also be applied to more standard formulations of the word-sense disambiguation task, we show how we can leverage the special properties of our formulation. We then present results for our algorithm, showing that it improves performance on this task. We also show how we can incorporate the results of the word translation task into a machine translation system.

2 Machine Translation

In machine translation, wish to translate a sentence $\mathbf{s}$ in our source language into $\mathbf{t}$ in our target language. The standard approach to machine translation uses the *source-channel model* ,

$$argmax_{\mathbf{t}} P(\mathbf{t}|\mathbf{s}) = argmax_{\mathbf{t}} P(\mathbf{t})P(\mathbf{s}|\mathbf{t}),$$

where $P(\mathbf{t})$ is the *language model* for the target language, and $P(\mathbf{s}|\mathbf{t})$ is an *alignment model* from the target language to the source language. Together they define a generative model for source/target pair $(\mathbf{s}, \mathbf{t})$: first $\mathbf{t}$ is generated according to the language model $P(\mathbf{t})$; then $\mathbf{s}$ is generated from $\mathbf{t}$ according to $P(\mathbf{s}|\mathbf{t})$.

Typically, strong independence assumptions are then made about the distribution $P(\mathbf{s}|\mathbf{t})$. For example, in the IBM Models (Brown et al., 1993), each word t_i independently generates 0, 1, or more words in the source language. Thus, the words generated by t_i are independent of the words generated by t_j for each $j \neq i$. This means that correlations between words in the source sentence are not captured by $P(\mathbf{s}|\mathbf{t})$, and so the type of features we use in our word translation models to predict t_i given s_i are not available to a system making these types of independence assumptions. In this type of system, semantic and syntactic relationships between words are only modeled in the target language; most or all of the semantic and syntactic information contained in the source sentence is ignored.

Standard n-gram language models assign probabilities to target sentences $\mathbf{t}$ by assigning probabilities to the component n-grams within the sentence. While some local context is captured by these models, the models are very sensitive to word order. Also, no longer range dependencies are captured by these models. Thus, language model also only captures a limited amount of semantic and syntactic information.

3 Task Formulation

We focus on determining a word or phrase in the target language t which is the translation for an individual word w in the source language s. Clearly, there are cases where w is part of a multi-word phrase that needs to be translated as a unit. Our approach can be extended to this case if we preprocess the data in s to find phrases, and then execute the entire algorithm treating phrases as atomic units. We do not explore this extension in this pa-

per, instead focusing on the word-to-phrase translation problem.

As we discussed, a key advantage of the word translation task vs. word sense disambiguation is the availability of large amounts of training data for machine translation. These data are in the form of bilingual corpora, such as the European Parliament proceedings[2]. Such documents provide many training instances, where a word in one language is translated into another. However, the data is only partially labeled in that we are not given a word-to-word alignment between the two languages, and thus we do not know what every word in the source language s translates to in the target language t. While sentence-to-sentence alignment is a fairly easy task, word-to-word alignment is considerably more difficult.

In order to obtain word-to-word alignments, we used GIZA++[3], an implementation of the IBM Models. As the goal is to obtain a large amount of data which will help us determine how to translate a particular word, we stemmed the text of both languages as a preprocessing step.

The alignment algorithm can be run in either direction. When run in the $s \rightarrow t$ direction, the algorithm aligns each word in s to at most one word in t. Consider some sentence in s that contains the word w, and let $u = t_1, \ldots, t_k$ be the set of words that align to w in the target language t. In general, we can consider u to be a candidate translation for w in t. However, this definition is quite noisy: a word t_i might have been aligned with w arbitrarily; or, t_i might be a word that itself corresponds to a multi-word translation in s. Thus, we also align the sentences in the $t \rightarrow s$ direction, and require that t_i aligns either with w or with nothing. We then say that w in s translates to a phrase $u = t_1, \ldots, t_k$ in t if w aligns with u in the $s \rightarrow t$ alignment, and each t_i lines up with either w or with nothing in the $t \rightarrow s$ direction. As this process is still fairly noisy, we only consider u to be a candidate translation for w only if it occurs some minimum number of times in the data.

The final result of our processing of the corpus is, for each source word w, a set of target words/phrases U_w, and a set of sentences where, in each sentence, w is aligned to some $u \in U_w$. For any sentence $\mathbf{w}$ and word w, let $u_{w,\mathbf{w}} \in U_w$ be the label assigned to w in $\mathbf{w}$. We can now treat this set of sentences as a fully-labeled corpus, which can be split into a set used for learning the word-translation model and a test set used for evaluating its performance.

We note, however, that there is a limitation to using accuracy on the test set for evaluating the performance of the algorithm. A source word w in a given context may have two equally good, interchangeable translations into the target language. Our evaluation metric only rewards the algorithm for selecting the target word/phrase that happened to be used in the actual translation. Thus, accuracies measured using this metric may be artificially low. This problem arises when defining an evaluation metric for any machine translation task. In our setting, we could correct for this problem either by using a thesaurus in the target language to identify synonyms, or (as done in the BLEU score (Papineni et al., 2002)) by considering multiple translations of the same sentence and accepting the translation associated with w in any of them. We return to this issue in Section 5.

4 Word Translation Algorithms

The word translation task and the word-sense disambiguation task have the same form: Each word w is associated with a set of possible labels U_w; given a sentence $\mathbf{w}$ containing word w, we must determine which of the possible labels in U_w to assign to w in the context $\mathbf{w}$. The only difference in the two tasks is the set U_w: for word translation it is set of possible translations of w, while for word sense disambiguation it is the set of possible senses of w in some ontology. Thus, we may use any word sense disambiguation algorithm as a word translation algorithm by appropriately defining the senses (assuming that the WSD algorithm does not assume that a particular ontology is used to choose the senses).

We considered two similar models for word translation: Naive Bayes, a generative model; and logistic regression (Minka, 2000), a discriminative model. While training for the Naive Bayes model is simple and efficient, for rich sets of features, the independence assumptions made in the

[2] Available at http://www.isi.edu/ koehn/

[3] Available at http://www.isi.edu/ och/GIZA++.html

German (freq.)	Translation
augenblick(24), moment(24)	instant, “in a minute”
minut redezeit(37)	minute speech
sitzungsprotokoll(11), protokoll sitzung(28)	meeting proceedings
parlament protokoll(181)	parlament proceedings
protokoll(1557)	proceedings
minut(1072), minut zeit(13)	a minute of time
minut lang(19)	minute-long
minut verfügung(22)	minute decision

Table 2: Aligned translations for ”meeting” occurring at least 10 times in the corpus

small; little; diminutive; minute; fine; inconsiderable; paltry; faint; slender; ... (25 words)
hour; day; week; month; quarter; year; decade; decenniumm lustrum; ... (18 words)
moment; instant; second; minute; twinkling; trice; flash; breath; crack; jiffy; ... (14 words)
minute; diminutive; microscopic; microzeal; inconsiderable; exiguous; puny; ... (32 words)
record; note; minute; register; registry; roll; cartulary; diptych; ... (25 words)
compendium; abstract; precis; epitome; multum in parvo; analysis; pandect; ... (21 words)

Table 3: Entries for ”minute” in Roget’s 1911 Thesaurus

model are highly violated. It is known both theoretically and empirically (e.g., (Ng and Jordan, 2002)) that discriminative models achieve higher accuracies than generative models if enough data is available. For the traditional word-sense disambiguation task, data must be hand-labeled, and is therefore usually too scarce to allow for discriminative training. In our setting, however, training data is acquired automatically from bilingual corpora, which are widely available. Thus, discriminative training is a viable option for the word translation problem.

4.1 Features

Our word translation model for a word w_i in a sentence $\mathbf{w} = w_1, \ldots, w_k$ is based on features constructed from the word and its context within the sentence:

- the part of speech of w_i (generated using the Brill tagger[4]);
- the close context of w_i — the (stemmed) words w_j for $|j - i| <= \delta$ for some small δ. Note that the since we do not consider the order of the words in the context of w_i, our features are more robust to word order than an n-gram model.

Let $\phi^{w_i,\mathbf{w}}$ be the set of features extracted for word w_i in the context of a sentence $\mathbf{w}$.

[4]Available at http://www.cs.jhu.edu/ brill/

4.2 Models

The Naive Bayes model encodes the joint distribution over the sentence and possible translations for each word w, $(P_w(u_{w,\mathbf{w}} = u, \mathbf{w}) : u \in U_w)$. It is parameterized by the conditional probability distributions $P_w(u_{w,\mathbf{w}})$, $P_w(w_i \mid u)$, and $P_w(pos(w) \mid u)$, where $pos(w)$ is the part of speech of w in sentence $\mathbf{w}$. Let $c_{\mathbf{w}}(w)$ be the nearby context of w in $\mathbf{w}$. Then we define $P_w(u, \mathbf{w})$ to be

$$P_w(u)P_w(pos(w) \mid u) \prod_{w' \in c_{\mathbf{w}}(w)} P_w(w' \mid u).$$

The logistic regression model instead encodes the conditional distribution $(P(u_{w,\mathbf{w}} = u \mid w, \mathbf{w}) : u \in U_w)$. Such a model is parameterized by a set of vectors $\boldsymbol{\theta}^w_u$, one for each word w and each possible target $u \in U_w$, where each vector contains a weight $\theta^w_{u,j}$ for each feature $\phi^{w,\mathbf{w}}_j$. We can now define our conditional distribution:

$$P_{\boldsymbol{\theta}^w}(u \mid w, \mathbf{w}) = \frac{1}{Z_{w,\mathbf{w}}} e^{\boldsymbol{\theta}^w_u \phi^{w,\mathbf{w}}}$$

where the partition function is

$$Z_{w,\mathbf{w}} = \sum_{u' \in U_w} e^{\boldsymbol{\theta}^w_{u'} \phi^{w,\mathbf{w}}}.$$

4.3 Single sense per discourse

We can use either of these models to classify each word w in each sentence $\mathbf{w}$ in isolation. However, this approach ignores an important source of information about the word sense: the observation that

many words (specifically nouns) tend to take only one of their possible senses (or translations) in a single discourse.

This type of constraint can be incorporated into our framework by considering the set of classification decisions for a given word w in a set of related sentences not as a set of independent classification tasks, but as a *collective classification* task. It is particularly natural to write this model as an extension of logistic regression, as follows.

We construct a Markov network (Pearl, 1988), with a node (variable) $Y_{\mathbf{w}}$ for each sentence $\mathbf{w}$ containing the word w in the discourse; the value of $Y_{\mathbf{w}}$ corresponds to the label of w in $\mathbf{w}$. Each variable $Y_{\mathbf{w}}$ is associated with a *node potential*, which is simply the unnormalized conditional distribution $Z_{w,\mathbf{w}} P_{\boldsymbol{\theta}^w}(u_{w,\mathbf{w}} \mid w, \mathbf{w})$. We also define a set of *edge potentials* τ_w, which connect the variables corresponding to consecutive mentions of w in the discourse. These potentials encode our preference for maintaining the same translation for the two mentions. The product of these two sets of potentials defines an (unnormalized) probability function over all occurrences of the word w within the current discourse; importantly, it can be written in an exponential form similar to logistic regression.

Extending Naive Bayes to use single sense per discourse is less straightforward; we extend Naive Bayes using a hybrid discriminative-generative model. Namely, we calculate the conditional distribution $P(\phi^{w_i,\mathbf{w}} | u_{w,\mathbf{w}})$ for each sentence $\mathbf{w}$ using the joint distribution defined by our Naive Bayes model, replace $P_w(u)$ with a node potential $Y_{\mathbf{w}}$, and incorporate the same edge potentials τ_w as in the logistic case. The product of these three types of factors again produces a joint probability model over occurrences of w within the current discourse. Then, for each occurrence of word w, we choose the value of $u_{w,\mathbf{w}}$ which maximizes the marginal probability of $u_{w,\mathbf{w}}$.

In both cases, each word w defines its own similarity potential τ_w. Thus, different words may have stronger or weaker correlations between consecutive occurrences. This potential is a full potential, i.e. for each pair $u, u' \in U_w$, we may a different value for $\tau_w(u, u')$. Thus, we can also capture correlations between related but different translations of our source word w.

4.4 Training

One of the main advantages of the Naive Bayes model is that training is simple and efficient. We train the model in order to maximize the joint probability of the observed labels and the features. More precisely, consider the model for word w, and let D_w be the set of all sentences in our training data containing w. Our goal in training the model for w is to maximize

$$\prod_{\mathbf{w} \in D_w} P_w(u_{w,\mathbf{w}}, \mathbf{w}).$$

We can maximize this expression by calculating counts over our observed data. For example, we estimate

$$P_w(w' \mid u) = \frac{\sum_{\mathbf{w}} 1[w \in \mathbf{w}, w' \in c_{\mathbf{w}}(w)]}{\sum_{\mathbf{w}} 1[w \in \mathbf{w}]}.$$

We train the logistic regression model to instead maximize the conditional likelihood of the observed labels given the features in our training set. Thus our goal in training the model for w is to maximize

$$\prod_{\mathbf{w} \in D_w} P_{\boldsymbol{\theta}^w}(\mathbf{w} \mid w, \mathbf{w}).$$

We maximize this objective by maximizing its logarithm (the log-conditional-likelihood) using conjugate gradient descent (Shewchuk, 1994).

We also learn the parameters for the single sense per discourse edges in slightly different ways for each model. In the case of logistic regression, we optimize the parameter vectors $\boldsymbol{\theta}^w$ and the edge potentials τ_w simultaneously, using conjugate gradient descent. For the Naive Bayes model, we first estimate $P_w(w_i \mid u)$ and $P_w(pos(w) \mid u)$ using empirical counts as for the standard Naive Bayes model; we are then left with a Markov network which we can again optimize using gradient descent.

4.5 Thesaurus-based smoothing

One additional improvement that arises naturally in our task is a form of smoothing. Consider two words w and w' that have synonymous senses; in many cases, these words will also have similar

possible translations. In such cases, we can reduce the sparsity of our data by sharing the data for both words.

Specifically, consider words w and w' that have senses that appear in the same entry in some thesaurus. Moreover, assume that there is a possible translation u that appears both in U_w and in $U_{w'}$. We then count every sentence $\mathbf{w}$ where w' has label $u_{w'} = u$ as a sentence where w is translated to $u_w = u$ (and vice versa). More precisely, we add such a sentence $\mathbf{w}$ to D_w, and use them as part of the training, taking $\phi^{w,\mathbf{w}}$ to be $\phi^{w',\mathbf{w}}$.

For example, the word "web" occurs paired with "netz" only 10 times in our entire corpus, however, "network" (listed as a synonym in our thesaurus) is paired with "web" over 1700 times, and both words probably have very similar contexts for this meaning.

For this component we used the freely-available 1911 edition of Roget's Thesaurus[5]. We chose to use this thesaurus instead of WordNet because the size of the thesaurus entries were much larger on average for Roget's Thesaurus, allowing us to apply this procedure more often.

5 Experimental Results

We tested our single word accuracy on the first 25 ambiguous words (those with more than one candidate translation) chosen from a randomly selected document. We compared our models to two different baselines: one that chooses the most common translation for the given word and a second that chooses the most common translation given the tagger-generated parts of speech.

Our most accurate model improves 7% over the basic baseline and 6% over the part of speech baseline. In general, logistic with single sense per discourse worked the best, although this didn't hold for every word. There was large variation in our models' improvement over baseline. Sometimes the grouping improved our accuracy about one percent, but often it significantly hurt it. The single sense per discourse edges consistantly showed a small improvement for most of the words. The Naive Bayes model generally had the same or slightly lower accuracy than the Logistic model.

On a few of the words, we achieved very large increases in accuracy. The word "minute" had a baseline accuracy of 59% including part of speech. The logistic model improved the accuracy to 78%, and adding single sense per discource edges increased the accuracy to 80%. Our improvement was not limited to nouns: the verb "rise" showed a 15% increase over baseline without part of speech and an 8% increase over baseline with part of speech.

Our accuracies are artificially low since in many cases a single word can be translated to many different words with the same meaning, if we translate "minutes" to "sitzungsprotokoll" while in our corpus it was translated to "protokoll sitzung", we are marked wrong when we have a correct translation. At the same time the accuracies are artificially inflated by the fact that we only consider cases where we can find an aligned word in the German corpus, so translations where a word is dropped or inserted into a compound word are not counted.

5.1 Impact on Machine Translation

Using our word translation system, we obtain predictions for the translation of each source word s_i. We would like to use these predictions, which take into account local and nonlocal dependencies in the source language, in order to improve the performance of machine translation. For a baseline system, we used the CMU-Cambridge toolkit[6] to construct a language model and the already mentioned GIZA++ for an alignment model. The final component is a decoder, which searches in the space of target sentences $\mathbf{t}$ for a sentence maximizing $P(\mathbf{t}|\mathbf{s})$. We used a greedy decoder (Germann et al., 2001), the isi-rewrite-decoder[7].

In order to use our word-translation model for translating the sentence $\mathbf{s}$, we need to obtain predictions for the words s_i. There are several types of words we do not train models for. First, we did not train models for stop words; we rely on the language model to choose appropriate translations for these words. Second, we do not train models for any word having 0 or 1 candidate translations (recall that we only consider a possible translation if it occurs some minimum number of times).

[5] http://wiretap.area.com/Gopher/Library/Classic/roget.txt

[6] Available at http://mi.eng.cam.ac.uk/ prc14/toolkit.html

[7] Available at http://www.isi.edu/licensed-sw/rewrite-decoder/

Model	Macro Average	Micro Average
Baseline	0.526	0.434
Baseline with Part of Speech	0.536	0.442
Logistic	0.559	0.493
Logistic with Single Sense	0.564	0.503
Naive Bayes with Single Sense	0.563	0.502
Naive Bayes with Grouping	0.546	0.489
Logistic with Grouping	0.547	0.489

Table 4: Aligned Word Prediction Accuracy

The isi-rewrite decoder provides a way to force the word s_i to be translated as a particular word t_j. Thus, a natural way to use our word translation system is to force each word s_i to be translated as $argmax_t P(t|s_i, \mathbf{s})$. The problem with this method is that while we choose the right translation more often than baseline, we do not take into account the language model when choosing the correct translation. When not using single sense per discourse, our word-translation model makes the assumption that the most likely translations of words s_i and s_j are independent given the source sentence $\mathbf{s}$ (single sense per discourse introduces correlations between the predictions of different occurrences of the same word). A further problem is that we do not consider phrase-translations, which ignores a case particularly common for German where several English words translate to a single compound German word. Still, forcing the translator to choose our model's best guess improves the BLEU score 7

In order to demonstrate potential improvements over the IBM translation model, we restricted our predictions to words where we had a large gain over baseline on the word translation task. Specifically, for each of the words "minute" and "rise", we built a test set consisting of sentences containing these words. We then forced the translations of these words only, allowing the language and alignment models to predict all other words. Our performance vs. the IBM Model was lower for "minute" and higher. This may be related to the fact that the translations of "minute" depends highly on adjacent words (and thus can be predicted by the language model), while for "rise", a verb, the directly adjacent words may not be as useful.

A second way to use our predictions is to rerank the n-best candidate translations generated by the decoder. Unfortunately, the isi-rewrite decoder uses greedy search to find a good translation, and thus does not have the capability to produce a useful list of candidate translations.

A final way to use our predictions would be to modify the alignment model $P(\mathbf{s}|\mathbf{t})$ on a sentence-by-sentence basis, using the confidences generated by our word-translation model. While the isi-rewrite decoder does not directly allow for sentence-by-sentence parameter modification, we can achieve the same effect by running the decoder separately for each sentence, using a different set of parameters each time. Notice, however, that the alignment model is actually in the wrong direction, so this method does not have a probabilistic interpretation. We do not yet have experimental results for this method.

6 Related Work

(Diab and Resnik, 2002) suggests using large bilingual corpora in order to improve performance on the word sense disambiguation. The main idea is that knowing a German word may help determine the meaning of the corresponding English word. They apply this intuition to the Senseval word disambiguation task by running off-the-shelf translators in order to produce translations which they can then use for disambiguation.

(Ng et al., 2003) addresses word sense disambiguation by manually annotating each sense in WordNet with its translation in the target language (Chinese), and then producing labeled examples using the IBM Models. They show that they can achieve comparable results by replacing hand-labeled examples with examples automatically extracted from a bilingual corpus.

(Koehn and Knight, 2003) focuses on the task of noun-phrase translation. They improve performance on the noun-phrase translation task, and show that they can use this to improve full translations. A key difference is that in predicting noun-phrase translations, they do not consider the con-

Translation Model	Random Sentence Score	”Rise” Sentences Score	”Minute” Sentences Score
Force Any Word	0.1112		
Force Best Word	0.1193	0.0809	0.1134
No Modification	0.1196	0.0787	0.1161

Table 5: Translation BLEU Scores

text of nouns. They present results which indicate that humans can accurately translate noun phrases without looking at the surrounding context. We contend that while this may be true, context may still be helpful for a (sub-human-level) machine translator.

7 Discussion and Conclusions

In this paper we have addressed the word-translation problem. By viewing word-sense disambiguation in the context of a larger task, we are able to obtain large amounts of training data and directly evaluate the usefulness of our system for a real-world task. The word-translation task is an important subtask for machine translation, since while producing syntacticly well-formed sentences is very important for machine translation, the most spectacular failures of machine translation systems often are a result of semantically incorrect word translations. We have shown that we improve over a baseline system (choosing the most common translation) which is difficult to outperform in the word sense disambiguation task. Also, we presented results which indicate that this increased accuracy can lead to improved machine translation.

The word translation model could be improved in a variety of ways. Leveraging the fact that discriminative models can incorporate highly correlated features, we could use more complex features such as features extracted from an automatically generated parse tree. We can extend our model to deal with translating phrases as well as individual words; also, we can evaluate our model on other language pairs. Our data sharing model could likely be improved by more carefully choosing what words to share data between.

This work suggests an important and interesting direction in improving machine translation systems. Standard systems use very simple syntactic transformation models and very simple language models. Improvements can be gained by introducing into the model some richer model of the sentence being translated. This is illustrated in (Charniak et al., 2003) where sentence syntax is used in order to build a better translator.

Discriminative models can particularly easily incorporate constraints on the target sentence using information from the source sentence. This type of model might potentially lead to a general machine translation system with efficient decoding and increased translation accuracy.

References

P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation. *Computational Linguistics*, 19(2):263–313.

E. Charniak, K. Knight, and K. Yamada. 2003. Syntax-based language models for machine translation. *Proceedings of MY Summit IX*.

M. Diab and P. Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. *Proceedings of ACL*, pages 255–262.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada. 2001. Fast decoding and optimal decoding for machine translation. *Proceedings of ACL 39*.

P. Koehn and K. Knight. 2003. Feature-rich statistical translation of noun phrases. *Proceedings of ACL*, pages 311–318.

T. Minka. 2000. Algorithms for maximum-likelihood logistic regression. http://lib.stat.cmu.edu/ minka/papers/logreg.html.

A. Ng and M. Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems 14*.

H. T. Ng, B. Wang, and Y. S. Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. *Proceedings of ACL*, pages 455–462.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *Proceedings of ACL*.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco.

J. Shewchuk. 1994. An introduction to the conjugate gradient method without the agonizing pain. http://www-2.cs.cmu.edu/ jrs/jrspapers.html.