Word-Sense Disambiguation for Machine Translation

David Vickrey **Luke Biewald** **Marc Teyssier** **Daphne Koller**
Department of Computer Science
Stanford University
Stanford, CA 94305-9010
{dvickrey,lukeb,teyssier,koller}@cs.stanford.edu

Abstract

In word sense disambiguation, a system attempts to determine the sense of a 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. These issues stem partly from the fact that the task is being treated in isolation from possible uses of automatically disambiguated data. In this paper, we consider the related task of word translation, where we wish to determine the correct translation of a word from context. We can use parallel language corpora as a large supply of partially labeled data for this task. We present algorithms for solving the word translation problem and demonstrate a significant improvement over a baseline system. We then show that the word-translation system can be used to improve performance on a simplified machine-translation task and can effectively and accurately prune the set of candidate translations for a word.

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* (including closely related senses such as "an attitude of admiration or esteem" and "a feeling of friendship and esteem"); 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 proved difficult to perform significantly better than the baseline heuristic of choosing the most common sense for each word.[1]

In this paper, we address 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. Perhaps the most direct and compelling application of a 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. 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 word in the source language.

We therefore consider the task of word-sense disambiguation for the purpose of machine translation. Rather than predicting the sense of a particular word a, we predict the possible translations of a into the

[1] See, for example, results of Senseval-3, available at http://www.senseval.org/senseval3

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 771–778, Vancouver, October 2005.

target language. We both train and evaluate the system on this task. This formulation of the word-sense disambiguation task, which we 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 discuss how training data for this task can be acquired automatically from bilingual corpora. We apply a standard learning algorithm for word-sense disambiguation to the word translation task, with several modifications which proved useful for this task.We present the results of our algorithm on word translation, showing that it significantly improves performance on this task. We also consider two simple methods for incorporating word translation into machine translation. First, we can use the output of our model to help a translation model choose better words; since general translation is a very noisy process, we present results on a simplified translation task. Second, we show that the output of our model can be used to prune candidate word sets for translation; this could be used to significantly speed up current translation systems.

2 Machine Translation

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

$$\text{argmax}_{\mathbf{t}} P(\mathbf{t}|\mathbf{s}) = \text{argmax}_{\mathbf{t}} P(\mathbf{t})P(\mathbf{s}|\mathbf{t}),$$

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 the 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 $P(\mathbf{s}|\mathbf{t})$.[2]

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 context we will use in our word translation models to predict t_i given s_i is not available to a system making these 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. The language model $P(\mathbf{t})$ does introduce some context-dependencies, but the standard n-gram model used in machine translation is too weak to provide a reasonable solution to the strong independence assumptions made by the alignment model.

3 Task Formulation

We define the word translation task as finding, for an individual word a in the source language $\mathcal{S}$, the correct translation, either a word or phrase, in the target language $\mathcal{T}$. Clearly, there are cases where a is part of a multi-word phrase that needs to be translated as a unit. Our approach could be extended by preprocessing the data in $\mathcal{S}$ to find phrases, and then executing the entire algorithm treating phrases as atomic units. We do not explore this extension in this paper, instead focusing on the word-to-phrase translation problem.

As we discussed, a key advantage of the word translation vs. word sense disambiguation is the availability of large amounts of training data. This data is in the form of bilingual corpora, such as the European Parliament proceedings[3]. 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 $\mathcal{S}$ translates to in the target language $\mathcal{T}$. While sentence-to-sentence alignment is a fairly easy task, word-to-word alignment is considerably more difficult. To obtain word-to-word alignments, we used GIZA++[4], an implementation of the IBM Models (specifically, we used the output of IBM Model 4). We did not perform stemming on either language, so as to preserve suffix information for our word translation system and the machine translation language model.

Let $D_{\mathcal{S}}$ be the set of sentences in the source lan-

[2] Note that we refer to **t** as the target sentence, even though in the source-channel model, **t** is the source sentence which goes through the channel model $P(\mathbf{s}|\mathbf{t})$ to produce the observed sentence **s**.

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

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

French (frequency)	Translation
montée(51)	going up
lève(10), lever(17)	standing up
hausse(58), augmenter(37), augmentation(150)	increase(number)
interviens(53)	to rise to speak
naissance(21), source(10)	to be created, arise
soulevé(10)	raising an issue

Table 1: Aligned translations for "rise" occurring at least 10 times in the corpus

guage and $D_{\mathcal{T}}$ the set of target language sentences.

The alignment algorithm can be run in either direction. When run in the $\mathcal{S} \rightarrow \mathcal{T}$ direction, the algorithm aligns each word in $\mathbf{t}$ to at most one word in $\mathbf{s}$. Consider some source sentence $\mathbf{s}$ that contains the word a, and let $U_{a,\mathbf{s}\rightarrow\mathbf{t}} = b_1, \ldots, b_k$ be the set of words that align to a in the aligned sentence $\mathbf{t}$. In general, we can consider $U_a = \{U_{a,\mathbf{s}\rightarrow\mathbf{t}}\}_{\mathbf{s}\in D_a}$ to be the candidate set of translations for a in $\mathcal{T}$, where D_a is the set of source language sentences containing a. However, this definition is quite noisy: a word b_i might have been aligned with a arbitrarily; or, b_i might be a word that itself corresponds to a multi-word translation in $\mathcal{S}$. Thus, we also align the sentences in the $\mathcal{T} \rightarrow \mathcal{S}$ direction, and require that each b_i in the phrase aligns either with a or with nothing. As this process is still fairly noisy, we only consider a word or phrase $b \in U_a$ to be a candidate translation for a if it occurs some minimum number of times in the data.

For example, Table 1 shows a possible candidate set for the English word "rise", with French as the target language. Note that this set can contain not only target words corresponding to different meanings of "rise" (the rows in the table) but also words which correspond to different grammatical forms in the target language corresponding to different parts of speech, verb tenses, etc. So, disambiguation in this case is both over senses and grammatical forms.

The final result of our processing of the corpus is, for each source word a, a set of target words/phrases U_a; and a set of sentences D_a where, in each sentence, a is aligned to some $b \in U_a$. For any sentence $\mathbf{s} \in D_a$, aligned to some target sentence $\mathbf{t}$, let $u_{a,\mathbf{s}} \in U_a$ be the word or phrase in $\mathbf{t}$ aligned with a. 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 a 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 is a common problem with evaluating machine translation systems.

Another issue is that we take as ground truth the alignments produced by GIZA++. This has two implications: first, our training data may be noisy since some alignments may be incorrect; and second, our test data may not be completely accurate. As mentioned above, we only consider possible translations which occur some minimum number of times; this removes many of the mistakes made by GIZA++. Even if the test set is not 100% reliable, though, improvement over baseline performance is indicative of the potential of a method.

4 Word Translation Algorithms

The word translation task and the word-sense disambiguation task have the same form: each word a is associated with a set of possible labels U_a; given a sentence $\mathbf{s}$ containing word a, we must determine which of the possible labels in U_a to assign to a in the context $\mathbf{s}$. The only difference in the two tasks is the set U_a: for word translation it is the set of possible translations of a, while for word sense disambiguation it is the set of possible senses of a 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).

Our main focus in this paper is to show that machine learning techniques are effective for the word translation task, and to demonstrate that we can use the output of our word translation system to improve performance on two machine-translation related tasks. We will therefore restrict our attention to a relatively simple model, logistic regression (Minka, 2000). There are several motivations for using this discriminative, probabilistic model. First, 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 often too scarce to allow for discriminative training. In our setting, however, training data is acquired automatically from bilingual corpora, which are widely available and quite large. Thus, discriminative training is a viable option for the word translation problem. A second

consideration is that, to effectively incorporate our system into a statistical machine translation system, we would like to produce not just a single prediction, but a list of confidence-rated possibilities. The optimization procedure of logistic regression attempts to produce a distribution over possible translations which accurately represents the confidence of the model for each translation. By contrast, a classical Naive Bayes model often assigns very low probabilities to all but the most likely translation. Other word-sense disambiguation models may not produce confidence measures at all.

Features. Our word translation model for a word a in a sentence $\mathbf{s} = w_1, \ldots, w_k$ is based on features constructed from the word and its context within the sentence. Our basic logistic regression model uses the following features, which correspond to the feature space for a standard Naive Bayes model:

- the part of speech of a (generated using the Brill tagger)[5];
- a binary "occurs" variable for each word which is 1 if that word is in a fixed context centered at a (c_r words to the right and c_l words to the left), and 0 otherwise.

We also consider an extension to this model, where instead of the fixed context features above, we use:

- for each direction $d \in \{l, r\}$ and each possible context size $c_d \in \{1, ..., C_d\}$, an "occurs" variable for each word.

This is a true generalization of the previous context features, since it contains features for all possible context sizes, not just one particular fixed size. This feature set is equivalent to having one feature for each word in each context position, except that it will have a different prior over parameters under standard L_2 regularization. This feature set allows our model to distinguish between very local (often syntactic) features and somewhat longer range features whose exact position is not as important.

Let $\phi^{a,\mathbf{s}}$ be the set of features for word a to be translated, with sentence context $\mathbf{s}$ (the description of the model does not depend on the particular feature set selected).

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

$$P_{\boldsymbol{\theta}^a}(b \mid a, \mathbf{s}) = \frac{1}{Z_{a,\mathbf{s}}} e^{\boldsymbol{\theta}^a_b \phi^{a,\mathbf{s}}}$$

with partition function $Z_{a,\mathbf{s}} = \sum_{b' \in U_a} \exp(\boldsymbol{\theta}^a_{b'} \phi^{a,\mathbf{s}})$.

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

$$\prod_{\mathbf{s} \in D_a} P_{\boldsymbol{\theta}^a}(u_{a,\mathbf{s}} \mid a, \mathbf{s}).$$

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

One important consideration when training using maximum likelihood is regularization of the parameters. In the case of logistic regression, the most common type of regularization is L_2 regularization; we then maximize

$$\prod_{b,j} \exp\left(-\frac{(\theta^a_{b,j})^2}{2\sigma^2}\right) \prod_{\mathbf{s} \in D_a} P_{\boldsymbol{\theta}^a}(u_{a,\mathbf{s}} \mid a, \mathbf{s}).$$

This penalizes the likelihood for the distance of each parameter $\theta^a_{b,j}$ from 0; it corresponds to a Gaussian prior on each parameter with variance σ^2.

5 Word Translation Results

For our word translation experiments we used the European Parliament proceedings corpus, which contains approximately 27 million words in each of English and French (as well as a number of other languages). We tested on a set of 1859 ambiguous words — specifically, all ambiguous words contained in the first document of the corpus. For each of these words, we found all instances of the word in the corpus and split these instances into training and test sets.

We tested four different models. The first, Baseline, always chooses the most common translation for the word; the second, Baseline with Part of Speech, uses tagger-generated parts of speech to choose the most common translation for the observed word/part-of-speech pair. The third model, Simple Logistic, is the logistic regression model with the simpler feature set, a context window of a fixed size. We selected the window size by evaluating accuracy for a variety of window sizes on 20 of the 1859 ambiguous words using a random train-test split. The window size which performed best on average extended one word to the left and

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

Model	Macro	Micro
Baseline	0.511	0.526
Baseline with Part of Speech	0.519	0.532
Simple Logistic	0.581	0.605
Logistic	0.596	0.620

Table 2: Average Word Translation Accuracy

two words to the right (larger windows generally resulted in overfitting). The fourth model, Logistic, is the logistic regression model with overlapping context windows; the maximum window size for this model was four words to the left and four words to the right. We selected the standard deviation σ^2 for the logistic models by trying different values on the same small subset of the ambiguous words. For the Simple Logistic model, the best value was $\sigma^2 = 1$; for the Logistic model, it was 0.35.

Table 2 shows results of these four models. The first column is macro-averaged over the 1859 words, that is, the accuracy for each word counts equally towards the average. The second column shows the micro-averaged accuracy, where each test example counts equally. We will focus on the micro-averaged results, since they correspond to overall accuracy.

The less accurate of our two models, Simple Logistic, improves around 8% over the simple baseline and 7% over the part-of-speech baseline on average. Our more complex logistic model, which is able to handle larger context sizes without significantly overfitting, improves accuracy by another 1.5%.

There was a great deal of variance from word to word in the performance of our models relative to baseline. For a few words, we achieved very large increases in accuracy. For instance, the noun "agenda" showed a 31.2% increase over both baselines. Similarly, the word "rise" (either a noun or a verb) had part-of-speech baseline accuracy of 27.9%. Our model increased the accuracy to 57.0%.

It is worth repeating that accuracies on this task are artificially low since in many cases a single word can be translated to many different words with the same meaning. At the same time, accuracies are artificially inflated by the fact that we only consider examples where we can find an aligned word in the French corpus, so translations where a word is dropped or translated as part of a compound word are not counted.

One disadvantage of the EuroParl corpus is that it is not "balanced" in terms of semantic content. It is not clear how this affects our results.

6 Blank-Filling Task

One of the most difficult parts of machine translation is decoding — finding the most likely translation according to some probability model. The difficulty arises from the enormous number of possible translated sentences. Existing decoders generally use either highly pruned search or greedy heuristic search. In either case, the quality of a translation can vary greatly from sentence to sentence. This variation is much higher than the improvement in "semantic" accuracy our model is attempting to achieve. Moreover, currently available decoders do not provide a natural way to incorporate the results of a word translation system. For example, Carpuat and Wu (2005) obtain negative results for two methods of incorporating the output of a word-sense disambiguation system into a machine translation system.

Thus, we instead used our word translation model for a simplified translation problem. We prepared a dataset as follows: for each occurrence of an ambiguous words in an English sentence in the first document of the Europarl corpus, we tried to determine what the correct translation for that word was in the corresponding French sentence. If we found one and exactly one possible translation for that word in the French sentence, we replaced that word with a "blank", and linked the English word to that blank. The final result was a set of 655 sentences with a total of 3018 blanks.

For example, the following English-French sentence pair contains the two ambiguous words *address* and *issue* and one possible translation for each, *examiner* and *question*:

- Therefore, the commission should *address* the *issue* once and for all.
- Par conséquent, la commission devra enfin *examiner* cette *question* particulière.

We replace the translations of the ambiguous words with blanks; we would like a decoder to replace the blanks with the correct translations:

- Par conséquent, la commission devra enfin [*address*] cette [*issue*] particulière.

An advantage of this task is that, for a given distribution $P(\mathbf{t}|\mathbf{s})$, we can easily write a decoder which exhaustively searches the entire solution space for the best answer (provided that there are not too many blanks and that $P(\mathbf{t}|\mathbf{s})$ is sufficiently "local" with respect to $\mathbf{t}$). Thus, we can be sure that it is the probability model, and not the decoder, which is determining the quality of the output. Also, we have removed most or all syntactic variability from the task,

Model	λ_{lm}	λ_{ga}	λ_{da}	λ_{wt}	Acc
Language Model only	1	0	0	0	0.749
Source-Channel	1	1	0	0	0.821
LM + GA + DA	1	0.6^*	0.6^*	0	0.833
LM + GA + DA + WT	1	0.6^*	0^*	1.2^*	0.846

Table 3: Blank-filling results. Weights marked with * have been optimized.

allowing us to better gauge whether we are choosing *semantically* correct translations.

Let (a_i, b_i) be the pairs of words corresponding to the blanks in sentence $\mathbf{t}$. Then the alignment model decomposes as a product of terms over these pairs, e.g. $P(\mathbf{s}|\mathbf{t}) \propto \prod_{(a_i,b_i)} P(a_i|b_i)$. Analogously, we extend the word translation model as $P_{wt}(\mathbf{t}|\mathbf{s}) \propto \prod_{(a_i,b_i)} P_{wt}(b_i|\mathbf{s}, a_i)$.

The source-channel model can be used directly to solve the blank filling task; the language model makes use of the French words surrounding each blank, while the alignment model guesses the appropriate translation based on the aligned English word. As we have mentioned, this model does not take full advantage of the context in the English sentence. Thus, we hope that incorporating the word translation model into the decoder will improve performance on this task.

Conversely, simply using the word translation model alone for the blank-filling task would not take advantage of the available French context. There are four probability distributions we might consider using: the language model $P_{lm}(\mathbf{t})$; the "generative" alignment model $P_{ga}(\mathbf{s}|\mathbf{t})$, which we calculate using the training samples from the previous section; the analogous "discriminative" alignment model $P_{da}(\mathbf{t}|\mathbf{s})$, which corresponds to the Baseline system we compared to on the word translation task; and our overlapping context logistic model, $P_{wt}(\mathbf{t}|\mathbf{s})$, which also goes in the "discriminative" direction, but uses the context features in the source language for determining the distribution over each word's possible translations.

We combine these models by simply taking a log-linear combination:

$$\log P(\mathbf{t}|\mathbf{s}) \propto \lambda_{lm} \log P_{lm}(\mathbf{t}) + \lambda_{ga} \log P_{ga}(\mathbf{s}|\mathbf{t}) + \lambda_{da} \log P_{da}(\mathbf{t}|\mathbf{s}) + \lambda_{wt} \log P_{wt}(\mathbf{t}|\mathbf{s}).$$

The case of $\lambda_{lm} = \lambda_{ga} = 1$ and $\lambda_{da} = \lambda_{wt} = 0$ reduces to the source-channel model; other settings incorporate discriminative models to varying degrees.

We evaluated this combined translation model on the blank-filling task for various settings of the mixture coefficients λ. For our language model we used the CMU-Cambridge toolkit.[6] The word translation model for each ambiguous word was trained on all documents except the first.

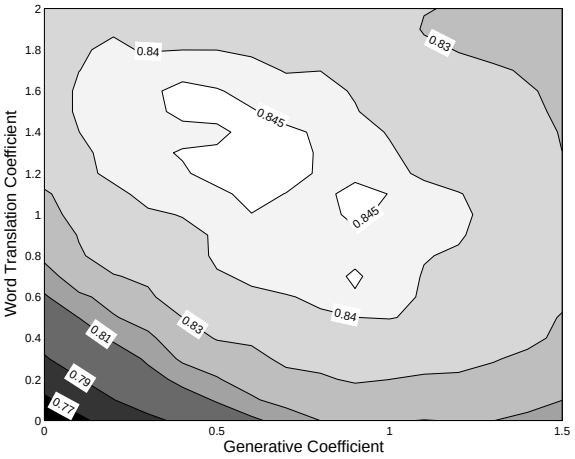

Figure 1: Accuracy on blank-filling task with $\lambda_{lm} = 1$ and $\lambda_{disc} = 0$ as a function of λ_{gen} and λ_{wt}.

Table 3 shows results for several sets of weights. A * denotes entries which we optimized (see below); other entries were fixed. For example, the third model was obtained by fixing the coefficient of the language model to 1 and the word-translation to 0, and optimizing the weights for the generative and discriminative alignment models.

The language model alone is able to achieve reasonable results; adding the alignment models improves performance further. By adding the word-translation model, we are able to improve performance by approximately 2.5% over the source-channel model, a relative error reduction of 14%, and 1.3% over the optimized model using the language model and generative and discriminative alignment models, a relative error reduction of 7.8%.

We chose optimal coefficients for the combined probability models by exhaustively trying all possible settings of the weights, at a resolution of 0.1, evaluating accuracy for each one on the test set. Figure 1 shows the performance on the blank-filling task as a function of the weights of the generative alignment model and the word-translation model (the optimum value of the discriminative alignment model $P(\mathbf{t}|\mathbf{s})$ is always 0 when we include the word-translation model). As we can see, the performance of this model is robust with respect to the exact value of the coefficients. The "obvious" setting of 1.0 for the generative model and 1.0 for the word translation model performs nearly as well

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

as the optimized setting. In the optimal region, the word-translation model receives twice as much weight as the generative alignment model, indicating that word-translation model is more informative than the generative alignment model. Incorporating the discriminative alignment model into the source-channel model also improves performance, but not nearly as much as using the word-translation model.

An alternate way to optimize weights over translation features is described in Och and Ney (2002). They consider a number of translation features, including the language model and generative and discriminative alignment models.

7 Search Space Pruning

As we have mentioned, one of the main difficulties in translation is that there are an enormous number of possible translations to consider. Decoding algorithms must therefore use some kind of search-space pruning in order to be efficient. A key part of pruning the search space is deciding on the set of words to consider in possible translations (Germann et al., 2001). One standard method is to consider only target words which have high probability according to the discriminative alignment model. But we have already shown that the word translation model achieves much better performance on word translation than this baseline model; thus, we would expect the word translation model to improve accuracy when used to pick sets of candidate translations.

Given a probability distribution over possible translations of a word, $P(b|a, \mathbf{s})$, there are several ways to choose a reduced set of possible translations. Two commonly used methods are to only consider the top n scoring words from this distribution (*best-n*); and to only consider words b such that $P(b|a, \mathbf{s})$ is above some fixed threshold (*cut-off*).

We use the same data set as for the blank-filling task. We evaluate the accuracy of a pruning strategy by evaluating whether the correct translation is in the candidate set selected by the pruning strategy. To compare results for different pruning strategies, we plot performance as a function of average size of the candidate translation set. Figure 2 shows the accuracy vs. average candidate set size for the word-translation model, discriminative alignment model, and generative alignment model.

The generative alignment model has the worst performance of the three. This is not surprising as it does not take into account the prior probability of the target word $P(b)$. More interestingly, we see that the word-translation model outperforms the discriminative translation model by a significant amount. For instance, to achieve 95% recall (that is, for 95% of the ambiguous words, we retain the correct translation), we only need candidate sets of average size 4.2 for the *cut-off* strategy using the word-translation model, whereas for the same strategy on the discriminative alignment model we require an average set size of 6.7 words.

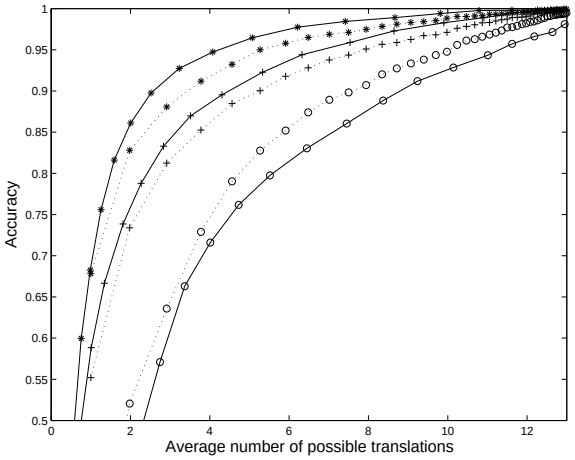

Figure 2: Accuracy of *best-n* strategy (dotted lines) and *cut-off* strategy (solid lines). o = generative alignment, + = discriminative alignment, * = word translation.

As the size of the solution space grows exponentially with the size of the candidate sets, the word-translation model could potentially greatly reduce the search space while maintaining good accuracy.

It would be interesting to use similar techniques to learn null fertility (i.e., when a word a has no translation in the target sentence $\mathbf{t}$).

8 Related Work

Berger et al. (1996) apply maximum entropy methods (equivalent to logistic regression) to, among other tasks, the word-translation task. However, no quantitative results are presented. In this paper we demonstrate that the method can improve performance on a large data set and show how it might be used to improve machine translation.

Diab and Resnik (2002) suggest using large bilingual corpora to improve performance on word sense disambiguation. The main idea is that knowing a French 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 to produce translations which they then use for disambiguation.

Ng et al. (2003) address word sense disambiguation by manually annotating WordNet senses with their translation in the target language (Chinese), and then automatically extracting labeled examples for word sense disambiguation by applying the IBM

Models to a bilingual corpus. They achieve comparable results to training on hand-labeled examples.

Koehn and Knight (2003) focus 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 context of nouns. They present results which indicate that humans can accurately translate noun phrases without looking at the surrounding context. However, as we have demonstrated in this paper, context can be very useful for a (sub-human-level) machine translator.

A similar argument applies to phrase-based translation methods (e.g., Koehn et al. (2003)). While phrase-based systems do take into account context within phrases, they are not able to use context across phrase boundaries. This is especially important when ambiguous words do not occur as part of a phrase — verbs in particular often appear alone.

9 Conclusions

In this paper, we focus on the word-translation problem. By viewing word-sense disambiguation in the context of a larger task, we were able to obtain large amounts of training data and directly evaluate the usefulness of our system for a real-world task. Our results improve over a baseline which is difficult to outperform in the word sense disambiguation task.

The word translation model could be improved in a variety of ways, drawing upon the large body of work on word-sense disambiguation. In particular, there are many types of context features which could be used to improve word translation performance, but which are not available to standard machine-translation systems. Also, the model could be extended to handle phrases.

To evaluate word translation in the context of a machine translation task, we introduce the novel blank-filling task, which decouples the impact of word translation from a variety of other factors, such as syntactic correctness. For this task, increased word-translation accuracy leads to improved machine translation. We also show that the word translation model is effective at choosing sets of candidate translations, suggesting that a word translation component would be immediately useful to current machine translations systems.

There are several ways in which the results of word translation could be integrated into a full translation system. Most naturally, the word translation model can be used directly to modify the score of different translations. Alternatively, a decoder can produce several candidate translations, which can be reranked using the word translation model. Unfortunately, we were unable to try these approaches, due to the lack of an appropriate publicly available decoder. Carpuat and Wu (2005) recently observed that simpler integration approaches, such as forcing the machine translation system to use the word translation model's first choice, do not improve translation results. Together, these results suggest that one should incorporate the results of word translation in a "soft" way, allowing the word translation, alignment, and language models to work together to produce coherent translations. Given an appropriate decoder, trying such a unified approach is straightforward, and would provide insight about the value of word translation.

References

A. Berger, S. Della Pietra, and V. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1).

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

M. Carpuat and D. Wu. 2005. Word sense disambiguation vs. statistical machine translation. *Proc. ACL*.

M. Diab and P. Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. *Proc. ACL*.

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. *Proc. ACL*.

P. Koehn and K. Knight. 2003. Feature-rich statistical translation of noun phrases. *Proc. ACL*.

P. Koehn, F. Och, and D. Marcu. 2003. Statistical phrase-based translation. *HLT/NAACL*.

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. *Proc. NIPS*.

H. T. Ng, B. Wang, and Y. S. Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. *Proc. ACL*.

F. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. *Proc. ACL*.

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