Improved Hidden Markov Models for Statistical Word Alignment

Adam Lopez
Institute for Advanced Computer Studies
Department of Computer Science
University of Maryland
College Park, MD 20742
alopez@cs.umd.edu

Philip Resnik
Institute for Advanced Computer Studies
Department of Linguistics
University of Maryland
College Park, MD 20742
resnik@umiacs.umd.edu

Abstract

We introduce several improvements to statistical word alignment based on the Hidden Markov Model. One improvement incorporates syntactic knowledge. Empirical results on a small corpus show that alignment performance exceeds that of a state-of-the-art system based on more complex models, implying that the method will be useful for low-density languages.1

1 Introduction

Word alignment is a vital subtask of many cross-lingual applications in natural language processing. It is a necessary first step in the parameter estimation methods of statistical translation models (Brown et al., 1993; Och et al., 1999; Koehn et al., 2003). It is used in automatic translation dictionary extraction from parallel corpora (Melamed, 1996). It can also be used as a bridge to port syntactic applications across languages (Yarowsky and Ngai, 2001; Smith and Smith, 2004).

The objective of a word alignment is to discover the word-to-word correspondences in a bilingual corpus of S sentence pairs, which we denote \{((f^{(s)})_i, (e^{(s)})_i) : s \in [1,S]\). Each sentence pair \((f, e) = (f^M, e^N)\) consists of a sentence \(f\) in one language and its translation \(e\) in the other, with lengths \(M\) and \(N\), respectively. Correspondences in a sentence consist of sets of links between words. A link \((f_j, e_i)\) indicates that the \(i\)th word \(e_i\) of \(e\) corresponds to – or is translated by – the \(j\)th word \(f_j\) of \(f\).

Alignment models come in two flavors: symmetric and asymmetric. Symmetric models often admit the entire \(O(2^{MN})\) search space. Although it is common to restrict this space – for instance, by allowing no word in either sentence to participate in more than one link, or by permitting multiple links only to contiguous words – the restriction always applies equally to both \(e\) and \(f\). Asymmetric models must be searched with heuristic methods (Melamed, 2000; Cherry and Lin, 2003).

Asymmetric models restrict the search space by requiring that each word in \(f\) participate in exactly one link. The behavior of words in \(e\) is unrestricted. We call this the asymmetry condition. If there are no dependencies between links, the search space reduces to a set of \(M\) decisions with \(N\) possible outcomes. This \(O(MN)\) space can be searched in its entirety. IBM Models 1 and 2 take this form (Brown et al., 1993). Limited dependencies can be introduced while maintaining a polynomial search space. The Hidden Markov Model (HMM) is an example of an asymmetric model with an \(O(M^2N)\) search space (Vogel et al., 1996). Asymmetric alignment models can be used to produce symmetric alignments if we apply the model twice – restricting each side in turn – and combine the results using simple methods based on intersection or union of resulting correspondence sets (Och et al., 1999; Koehn et al., 2003).

The most widely used alignment model is IBM Model 4 (Brown et al., 1993), which is a component of several of the previously mentioned applications (Och et al., 1999; Yarowsky and Ngai, 2001; Koehn et al., 2003; Smith and Smith, 2004). Its widespread use is partly due to freely available implementations (Al-Onaizan et al., 1999; Och and Ney, 2003), as

---

1Submitted for publication. Do not distribute or cite.
well as empirical studies which validate its performance over the other IBM Models and HMM Models (Och and Ney, 2003). This model also performed well in a recent comparison of several alignment systems (Mihalcea and Pedersen, 2003). On some test cases it yielded the best performance.

IBM Model 4 is an asymmetric exponential alignment model; it observes the asymmetry condition but its search space cannot be efficiently enumerated due to a complex set of dependencies between alignment links. In implementations of IBM Model 4, an asymmetric polynomial model, such as IBM Model 2 or an HMM Model, is used to generate an initial alignment which is then improved using a hill-climbing approach (Brown et al., 1993; Och and Ney, 2003). Although IBM Model 4 yields good results, we observe that its performance is only slightly better than the HMM model used to bootstrap it (Figure 1). The relatively small difference in performance between IBM Model 4 and the underlying HMM Model has been observed before Och and Ney (2003).

Based on this understanding of the IBM Model 4 search procedure and the observation of its performance in these trials, we hypothesize that implementations of IBM Model 4 derive most of their performance from the underlying HMM Model. Furthermore, owing to the simplicity of HMM Models, we believe that they are more conducive to study and improvement than exponential models such as IBM Model 4. We illustrate this point with several improvements to an HMM model which yield positive results in an empirical trial.

2 HMMs and Word Alignment

The HMM is a simple statistical sequence model in which each element $o_i$ of an observed sequence $o = o_1^M$ is emitted by a corresponding hidden state $q_i \in [1,N]$, according to a distribution $p(o_i|q_i)$. The hidden state sequence $q = q_1^M$ is generated by a simple bigram distribution $p(q_i|q_{i-1})$ of transitions between states. Thus the probability for an entire sequence is given by Equation 1.

$$P(o,q) = \prod_{i=1}^{M} p(q_i|q_{i-1}) \cdot p(o_i|q_i)$$  \hspace{1cm} (1)

When the state sequence is unknown we simply sum over all possible state sequences. This model is applicable to numerous problems in natural language processing, such as part-of-speech tagging (Merialdo, 1994). A key benefit of HMMs is the availability of standard algorithms for parameter estimation (Baum, 1972) and maximization (Viterbi, 1967).

Alignment models often arise out of more general models of correspondence. These models attempt to capture a joint model of sentence distribution, $P(f,e)$. This can be decomposed as in Equation 2, also known as “The Fundamental Equation of Machine Translation” (Brown et al., 1993).

$$P(f,e) = P(e) \cdot P(f|e)$$ \hspace{1cm} (2)

This decomposition corresponds to a process whereby $e$ is first produced according to the distribution $P(e)$, and $f$ is then produced according to the distribution $P(f|e)$. We call $P(e)$ the language model and $P(f|e)$ the translation model.
3 Changes to the Basic Model

3.1 Syntactic Distortion Parameters and Combined Distortion Models

Although Equation 3 is adequate in practice, it is not ideal. Numerous parameterizations have been proposed for the distortion probability. A good parameterization is to allow it to depend only on the distortion distance and an automatically determined word class \( C(e_{a_{i-1}}) \) as shown in Equation 4 (Och and Ney, 2000).

\[
d(a_{i}|a_{i-1}) = p(a_{i}|a_{i} - a_{i-1}, C(e_{a_{i-1}}))
\]

This parameterization is subject to the criticism of many statistical translation models: it takes an impoverished view of the rich structure of language in favor of surface approximations. Although it can capture local movement, it cannot capture movement of whole structures or the behavior of long-distance dependencies across translations. The intuitive appeal of capturing this type of rich information has inspired numerous alignment models (Wu, 1995; Yamada and Knight, 2001; Cherry and Lin, 2003). Syntactic constraints are thought to offer several hints to an alignment model.

(1) Knowledge of a word’s syntactic collocates could be used to influence the translation choice of the word.

(2) Non-overlapping syntactic units usually do not overlap in translation. This is known as phrase cohesion (Fox, 2002; Cherry and Lin, 2003).

(3) Words local to the same syntactic unit tend to remain local to each other across translation. This is the inverse of (2).

In support of this view, empirical studies have shown that syntactic phrases tend to move as a unit in translation (Fox, 2002) and that significant syntactic information is preserved in translation (Hwa et al., 2002).

We might be able to express (1) in our HMM using a Maximum Entropy approach, but this would come at the expense of significant additional computation (Varea et al., 2002). We don’t know of any way to express (2), but we can express (3). We introduce a distortion model which depends on the tree

---

3 We ignore the sentence length probability \( p(M|N) \), which is not relevant to word alignment.
distance between two words. Given a dependency parse of \( e_i^M \), we define the tree distance \( \tau(e_i, e_k) \) between two English words \( e_i \) and \( e_k \) to consist of three components \((w, x, y)\). The first element \( w = \{1 \text{ if } i > k; 0 \text{ otherwise} \} \) is simply a binary indicator of the linear relationship of the words within the surface string. The variables \( x \) and \( y \) represent the respective number of dependency links separating \( e_i \) and \( e_k \) from a common ancestor node in the parse tree. The tree distance is effectively a measure of relationship. For instance, \( \tau(e_i, e_k) = (1, 1, 0) \) would indicate that \( e_k \) appears to the left of \( e_i \) and is also \( e_i \)’s parent in the dependency graph, while \( \tau(e_i, e_k) = (0, 1, 1) \) indicates that \( e_k \) appears to the right of \( e_i \) and is \( e_i \)’s sibling in the graph. In our tree distortion model, we condition on the tree distance and the part of speech \( T(e_{i-1}) \), giving us Equation 5.

\[
d(a_i|a_{i-1}) = p(a_i|\tau(e_{a_i}, e_{a_{i-1}}), T(e_{a_{i-1}})) \tag{5}
\]

The tree distance could easily be adapted to work with phrase-structure parses or tree-adjoining parses instead of dependency parses. For our purposes, the type and accuracy of parser are less important than consistency. In our experiments, the dependency parse and parts of speech are induced by mini-pair (Lin, 1998). This parser has been used in a much different asymmetric alignment model (Cherry and Lin, 2003).

In the absence of a parse, we could emulate the behavior of a surface distortion model with this model simply by supplying a parse in which each word modifies its predecessor in the sentence. This observation captures the main intuition of the tree distortion model – that it is simply an alternate view of the word order in a sentence.

We found that tree distortion works almost as well as surface distortion. However, a benefit of the HMM framework is that we do not need to commit to one view or analysis of the data. Since both of these distributions represent attempts to model \( p(a_i|a_{i-1}) \), we can combine them using linear interpolation as in Equation 6.

\[
d(a_i|a_{i-1}) = 
\lambda_{C(e_{a_{i-1}}), T(e_{a_{i-1}})} p(a_i|\tau(e_{a_i}, e_{a_{i-1}}), T(e_{a_{i-1}})) + 
(1 - \lambda_{C(e_{a_{i-1}}), T(e_{a_{i-1}})}) p(a_i|a_i - a_{i-1}, C(e_{a_{i-1}})) \tag{6}
\]

In principle, any number of alternative distortion models could be combined with this framework.

The distortion parameters are normalized for each sentence, as in Och and Ney (2000). In our experiments, we initialized the \( \lambda_{C,T} \) parameters to 0.5 and trained them with the other model parameters using Expectation-Maximization. This yielded a 1.5% improvement in AER over either model in isolation. Since we only had parses for English, we did not use tree distortion in the application of \( P(e|f) \), needed for symmetrization.

3.2 The Importance of Normalization: Fixing the Garbage Collection Problem

A frequently observed problem in many alignment models is the so-called “garbage collection” problem. This problem occurs when an infrequent word occurring in \( e \) is aligned to numerous words in \( f \). Moore (2004) presents a solution to this problem for IBM Model 1, which we incorporate into our initialization procedure as described in Section 3.3.

We observe a very simple reason for garbage collection in HMM methods. If we use \( t(f_i|e_a) = p(f_i|e_a) \) for our translation probability, the total probability mass for a common word \( e_i \) producing any of the \( f_1^M \) will most likely be less than the total probability mass for a rare word \( e_k \) producing any of the \( f_1^M \), even if the distribution of \( e_i \) is sharper than that of \( e_k \). We can solve this problem simply by normalizing the translation probabilities as follows:

\[
t(f_i|e_a) = \frac{p(f_i|e_a)}{\sum_{k=1}^{M} p(f_k|e_a)}
\]

This allows common words which distribute their translation probability mass across many words in many sentences to compete fairly with rare words, for which translation probability mass is concentrated on a small set of words within a few sentences. It also corresponds nicely to the normalization of the distortion parameters.
3.3 Improving Initialization

Our HMM will produce acceptable results if we initialize both emission and transition probabilities with a uniform distribution or IBM Model 1 translation probabilities. However, we can do better. We use the probabilities generated by an improved Model 1 based on log-likelihood ratios (Moore, 2004).

If we can use an improved initial estimate for the translation probabilities, we can also use one for distortion. Consider a simple distortion model \( p(a_i | a_i - a_{i-1}) \). We know intuitively that this parameter will approach its maximum when the absolute value of the distortion \( |a_i - a_{i-1}| \) is near 0, because we know that words tend to retain locality across translation. In fact, we can see that if the distribution is trained using Expectation Maximization, this convergence is nearly guaranteed, simply because there are more possible distortions of length 1 than of length 20 in any corpus. Rather than wait for this to occur, we initialize the distortion parameters to \( |a_i - a_{i-1}|^{1/2} / Z, \alpha < 0 \). We optimized \( \alpha \) on a development set, and chose \( Z \) so that distortion probabilities for each position sum to 1. We use 1 for the numerator of this quantity for a distortion distance of 0 – this causes the distortions of size -1, 0, or 1 to have the highest initial probability. Although we actually use a more complex distortion model based on word classes, part of speech, and tree distance in our experiments, we initialize the distortion probabilities in the first HMM iteration from this simpler model.

3.4 Does NULL Matter in Asymmetric Alignment?

Och and Ney (2000) introduce a NULL-alignment capability to the HMM alignment model. This allows any word \( f_j \) to link to a special NULL word – by convention denoted \( e_0 \) – instead of one of the words \( e_i^N \). A link \((f_j, e_0)\) means that \( f_j \) does not correspond to any word in \( e \). This capability improved the performance of the HMM alignment in the absence of symmetrization, presumably because it allows the model to be conservative when evidence for an alignment is lacking.

We hypothesize that NULL alignment is unnecessary for asymmetric alignment models since we are going to intersect the resulting set of alignment links with a set of links from a complementary asymmetric set (Och et al., 1999; Koehn et al., 2003). This makes intuitive sense: if we don’t permit NULL alignments, then we expect to produce an alignment with high recall, but low precision; the intersection of two alignments should result in a high recall, high precision alignment. If we allow NULL alignments, we may produce a high-precision, low-recall initial alignment, but the intersection of this alignment will only result in further impacts on recall. This hypothesis is confirmed in our results.

Figure 2: Comparison of Alignment Error Rates between the two methods, using forward, backward, and combined methods of computing the alignment.

4 Experiments

We trained an HMM model incorporating our modifications on a very small portion of the Canadian Hansards, consisting of 5000 sentences. Results were calculated by extracting the resulting alignment on a set of test sentences and comparing with a manually annotated alignment. The training, test, and development sentences were the same as those used in Mihalcea and Pedersen (2003). We measured the accuracy of the error rate using the standard Alignment Error Rate (AER) metric, which combines recall over a set of sure alignments and precision over a set of probable alignments (Och and Ney, 2000).

Our results are shown in Figure 2. We observed that the improved HMM alignment model achieved
the best results after only four iterations of training, with an additional one iteration of training the Moore Model 1. After this point, performance tended to decline. By way of contrast, the training of the Model 4 sequence took about twice as many iterations to converge, but alignment error rate tended to stabilize after this point. Perplexity of both models continued to decrease, suggesting that the HMM fits the training data more quickly, but tends to overfit. Overall, the best Alignment Error Rate achieved by any iteration of the HMM model was 12.76%, compared with 18.76% for IBM Model 4. By way of comparison, the best Model 4 alignment error rates for 50K and 100K sentences were 12.1% and 11.34%, respectively.

5 Conclusions and Future Work

We have illustrated the utility of the HMM model for word alignment by introducing several novel improvements. These improvements indicate that this largely overlooked model is an excellent platform for research in word alignment. Our improvements yield good results on a small corpus, indicating the method’s usefulness for low-density languages. We plan to investigate this further.

Although we have consciously avoided using annotation in both languages due to this interest, this method could easily be extended to use annotations in both languages, as in Toutanova et al. (2002).

We also plan to perform extrinsic evaluations by using the word alignments in the parameter estimation method of a phrase-based statistical machine translation system (Och et al., 1999; Koehn et al., 2003).

References


