The Duluth Word Alignment System

Bridget Thomson McInnes
Department of Computer Science
University of Minnesota
Duluth, MN 55812
bthomson@d.umn.edu

Ted Pedersen
Department of Computer Science
University of Minnesota
Duluth, MN 55812
tpederse@umn.edu

Abstract

The Duluth Word Alignment System participated in the 2003 HLT-NAACL Workshop on Parallel Text shared task on word alignment for both English–French and Romanian–English. It is a Perl implementation of IBM Model 2. We used approximately 50,000 aligned sentences as training data for each language pair, and found the results for Romanian–English to be somewhat better. We also varied the Model 2 distortion parameters among the values 2, 4, and 6, but did not observe any significant differences in performance as a result.

1 Introduction

Word alignment is a crucial part of any Machine Translation system, since it is the process of determining which words in a given source and target language sentence pair are translations of each other. This is a token level task, meaning that each word (token) in the source text is aligned with its corresponding translation in the target text.

The Duluth Word Alignment System is a Perl implementation of IBM Model 2 (Brown et al., 1993). It learns a probabilistic model from sentence aligned parallel text that can then be used to align the words in another such text (that was not a part of the training process).

A parallel text consists of a source language text and its translation into some target language. If we have determined which sentences are translations of each other then the text is said to be sentence aligned, where we call a source and target language sentence that are translations of each other a *sentence pair*.

(Brown et al., 1993) introduced five statistical translation models (IBM Models 1 – 5). In general a statistical machine translation system is composed of three components: a language model, a translation model, and a decoder (Brown et al., 1988).

The language model tells how probable a given sentence is in the source language, the translation model indicates how likely it is that a particular target sentence is a translation of a given source sentence, and the decoder is what actually takes a source sentence as input and produces its translation as output. Our focus is on translation models, since that is where word alignment is carried out.

The IBM Models start very simply and grow steadily more complex. IBM Model 1 is based solely on the probability that a given word in the source language translates as a particular word in the target language. Thus, a word in the first position of the source sentence is just as likely to translate to a word in the target sentence that is in the first position versus one at the last position. IBM Model 2 augments these translation probabilities by taking into account how likely it is for words at particular positions in a sentence pair to be alignments of each other.

This paper continues with a more detailed description of IBM Model 2. It goes on to present the implementation details of the Duluth Word Alignment System. Then we describe the data and the parameters that were used during the training and testing stages of the shared task on word alignment. Finally, we discuss our experimental results and briefly outline our future plans.

2 IBM Model 2

Model 2 is trained with sentence aligned parallel corpora. However, our goal is learn a model that can perform word alignment, and there are no examples of word alignments given in the training data. Thus, we must cast the training process as a missing data problem, where we learn about word alignments from corpora where only sentence (but not word) alignments are available. As is common with missing data problems, we use the Expectation–Maximization (EM) Algorithm (Dempster et al., 1977) to estimate the probabilities of word alignments in this model.

The objective of Model 2 is to estimate the probability that a given sentence pair is aligned a certain way. This is represented by $P(a|s,t)$, where s is the source sentence, t is the target sentence, and a is the proposed word alignment for the sentence pair. However, since this probability can't be estimated directly from the training data, we must reformulate it so we can use the EM algorithm. From Bayes Rule we arrive at:

$$P(a|s,t) = \frac{P(a,t|s)}{\sum_a P(a,t|s)}, \qquad (1)$$

where $P(a,t|s)$ is the probability of a proposed alignment of the words in the target sentence to the words in the given source sentence. To estimate a probability for a particular alignment, we must estimate the numerator and then divide it by the sum of the probabilities of all possible alignments given the source sentence.

While clear in principle, there are usually a huge number of possible word alignments between a source and target sentence, so we can't simply estimate this for every possible alignment. Model 2 incorporates a *distortion factor* to limit the number of possible alignments that are considered. This factor defines the number of positions a source word may move when it is translated into the target sentence. For example, given a distortion factor of two, a source word could align with a word up to two positions to the left or right of the corresponding target word's position.

Model 2 is based on the probability of a source and target word being translations of each other, and the probability that words at particular source and target positions are translations of each other (without regard to what those words are). Thus, the numerator in Equation 1 is estimated as follows:

$$P(a,t|s) = \prod_{j=1}^{m} t(t_j|s_{a_j}) \times a(a_j|j,l,m), \qquad (2)$$

The translation probability, $t(t_j|s_{a_j})$, is the likelihood that t_j, the target word at position j, is the translation of a given source word s that occurs at position a_j. The alignment probability, $a(a_j|j,l,m)$, is the likelihood that position a_j in the source sentence can align to a given position j in the target sentence, where l and m are the given lengths of the source and target sentences.

The denominator in Equation 1 is the sum of all the probabilities of all the possible alignments of a sentence pair. This can be estimated by taking the product of the sums of the translational and positional alignment probabilities.

$$\sum_a P(a,t|s) = \prod_{j=1}^{m} \sum_{i=0}^{l} t(t_j|s_i) * a(i|j,l,m), \qquad (3)$$

where i represents a position in the source sentence and all the other terms are as described previously.

The EM algorithm begins by randomly initializing the translation and positional alignment probabilities in Equation 2. Then it estimates Equation 3 based on these values, which are then maximized for all the target words according to Equation 1. The re–estimated translation and positional alignment probabilities are normalized and the EM algorithm repeats the above process for a predetermined number of iterations or until it converges.

3 System Components

The Duluth Word Alignment System consists of two preprocessing programs (plain2snt and snt2matrix) and one that learns the word alignment model (model2). These are all implemented in Perl.

The plain2snt program converts raw sentence aligned parallel text into the *snt* format, where each word type in the source and target text is represented as a unique integer. This program also outputs two vocabulary files for the source and target languages that list the word types and their integer values. This is closely modeled after what is done in the GIZA++ tool kit (Och and Ney, 2000b).

The snt2matrix program takes the snt file from plain2snt as input and outputs two files. The first is an adjacency list of possible word translations for each sentence pair. The second file consists of a table of alignment positions that were observed in the training corpora. The value of the distortion factor determines which positions may be aligned with each other.

The program model2 implements IBM Model 2 as discussed in the previous section. This program requires the vocabulary files, the snt file, the alignment positional probability file and the adjacency list file created by the plain2snt and snt2matrix programs. This program carries out the EM algorithm and estimates the probability of an alignment given the source and target sentences from the snt file. The model2 program outputs a file of word alignments for each of the training sentences and two files containing estimated values for word translation and positional alignment probabilities. Finally, there is also a program (test) that word aligns parallel text based on the output of the model2 program.

4 Experimental Framework

The Duluth Word Alignment System participated in both the English–French (UMD-EF) and Romanian–English (UMD-RE) portions of the shared task on word alignment.

The UMD-RE models were trained using 49,284 sentence pairs of Romanian-English, which was the complete set of training data as provided by the shared task organizers. It is made up of three different types of text: the novel *1984*, by George Orwell, which contains 6,429 sentence pairs, the Romanian Constitution which contains 967 sentence pairs, and a set of selected newspaper articles collected from the Internet that contain 41,889 sentences pairs. The gold standard data used in the shared task consists of 248 manually word aligned sentence pairs that were held out of the training process.

The UMD-EF models were trained using a 5% subset of the Aligned *Hansards* of the 36th Parliament of Canada (*Hansards*). The *Hansards* contains 1,254,001 sentence pairs, which is well beyond the quantity of data that our current system can train with. UMD-EF is trained on a balanced mixture of House and Senate debates and contains 49,393 sentence pairs. The gold standard data used in the shared task consists of 447 manually word aligned sentence pairs that were held out of the training process.

The UMD-RE and UMD-EF models were trained for thirty iterations. Three different models for each language pair were trained. These were based on distortion factors of two, four, and six. The resulting models will be referred to as UMD-XX-2, UMD-XX-4 and UMD-XX-6, where 2, 4, and 6 are the distortion factor and XX is the language pair (either RE or EF).

5 Experimental Results

The shared task allowed for two different types of alignments, Sure and Probable. As their names suggest, a sure alignment is one that is judged to be very likely, while a probable is somewhat less certain. The English–French gold standard data included S and P alignments, but our system does not make this distinction, and only outputs S alignments.

Submissions to the shared task evaluation were scored using precision, recall, the F-measure and the alignment error rate (AER). Precision is the number of correct alignments (C) out of the total number of alignments attempted by the system (S), while recall is the number of correct alignments (C) out of the total number of correct alignments (A) as given in the gold standard. That is,

$$precision = \frac{|C|}{|S|} \quad recall = \frac{|C|}{|A|} \tag{4}$$

The F–measure is the harmonic mean of precision and recall:

$$F\text{–}measure = \frac{2 \times precision \times recall}{precision + recall} \tag{5}$$

AER is defined by (Och and Ney, 2000a) and accounts for both Sure and Probable alignments in scoring.

The word alignment results attained by our models are shown in Table 1. We score and report our results as no-null, since our system does not include null alignments (source words that don't have a target translation). We also score relative to sure alignments only. During the shared task systems were scored with and without null alignments in the gold standard, so our results correspond to those without.

model	precision	recall	F-measure	AER
UMD-RE-2	.5292	.4706	.4982	.5018
UMD-RE-4	.5454	.4850	.5134	.4866
UMD-RE-6	.5352	.4745	.5030	.4970
UMD-EF-2	.5305	.2136	.3045	.4400
UMD-EF-4	.5422	.2183	.3112	.4279
UMD-EF-6	.5483	.2207	.3147	.4192

Table 1: No-null Alignment Results

It is apparent from Table 1 that the precision and recall of the models were not significantly affected by the distortion factor. Also, we note that the precision of the two language pairs is relatively similar. This may reflect that fact that we used approximately the same amount of training data for each language pair. However, note that the recall for English–French is much lower. We continue to investigate why this might be the case, but believe it may be due to the fact that the training data we randomly selected for the Hansards may not have been representative of the gold standard data.

Finally, the alignment error rate (AER) is lower (and hence better) for English–French than Romanian–English. However, note that the F–measure for Romanian–English is higher (and therefore better) than English–French. While this may seem contradictory, AER factors in both Sure and Probable alignments into is scoring while only the English–French data included such alignments in its gold standard.

The models used for our official submission to the shared task led to somewhat puzzling results, since as the number of iterations increased the precision and recall continued to fall. Upon further investigation, an error was found. Rather than estimating as shown in Equation 1, our system did the following:

$$P(a|s,t) = \frac{P(a,t|s)}{\prod\limits_{a} \sum\limits_{a} P(a,t|s)} \tag{6}$$

The results shown in Table 1 are based on a corrected version of the model. Thereafter as the number of iterations increased the accuracy of the results rose and then reached a plateau that was near what is reported here.

Table 2 includes the official results as submitted to the shared task based on the flawed model. These are designated as UMD.EF.1, UMD.RE.1, and UMD.RE.2. These use distortion parameters of 2 or 4, and were only trained for 4 iterations. However, it should be noted that the

model	precision	recall	F-measure	AER
UMD.RE.1	.5767	.4970	.5339	.4661
UMD.RE.2	.5829	.4999	.5382	.4618
UMD.EF.1	.3798	.6466	.4785	.3847

Table 2: No-null Alignment Results (original)

results are actually slightly better with respect to the F–measure and AER than our newer results.

6 Future Work

The mystery of why our flawed implementation of Model 2 performed better in some respects than our later repaired version is our current focus of attention. First, we must determine if our corrected Model 2 is really correct, and we are in the process of comparing it with existing implementations, most notably GIZA++. Second, we believe that the relatively small amount of training data might account for the somewhat unpredictable nature of these results. We will do experiments with larger amounts of training data to see if our new implementation improves.

However, we are currently unable to train our models in a practical amount of time (and memory) when there are more than 100,000 sentence pairs available. Clearly it is necessary to train on larger amounts of data, so we will be improving our implementation to make this possible. We are considering storing intermediate computations in a database such as Berkeley DB or NDBM in order to reduce the amount of memory our system consumes. We are also considering re–implementing our algorithms in the Perl Data Language (http://pdl.perl.org) which is a Perl module that is optimized for matrix and scientific computing.

Our ultimate objective is to extend the model such that it incorporates prior information about cognates or proper nouns that are not translated. Having this information included in the translation probabilities would provide reliable anchors around which other parameter estimates could be made.

Finally, having now had some experience with IBM Models 1 and 2, we we will continue on to explore IBM Model 3. In addition, we will do further studies with Models 1 and 2 and compare the impact of distortion factors as we experiment with different amounts of training data and different languages.

7 Acknowledgments

This system is being implemented by Bridget Thomson McInnes as a part of her M.S. thesis in the Department of Computer Science at the University of Minnesota, Duluth. The Perl code described in this paper is freely available from the authors.

This project has often been guided by an unpublished manuscript by Kevin Knight called *A Statistical MT Tutorial Workbook*. It's friendly tone helped keep this fun, at least most of the time.

References

P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin. 1988. A statistical approach to machine translation. In *Proceedings of the 12th International Conference on Computational Linguistics*.

P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

A. Dempster, N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–38.

F. Och and H. Ney. 2000a. A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th International Conference on Computational Linguistics*.

F. Och and H. Ney. 2000b. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*.