

Boosting Statistical Word Alignment

WU Hua, WANG Haifeng

Toshiba (China) Research and Development Center
5/F., Tower W2, Oriental Plaza
No.1, East Chang An Ave., Dong Cheng District
Beijing, 100738, China
{wuhua, wanghaifeng}@rdc.toshiba.com.cn

Abstract

This paper proposes an approach to improve statistical word alignment with the boosting method. Applying boosting to word alignment must solve two problems. The first is how to build the reference set for the training data. We propose an approach to automatically build a pseudo reference set, which can avoid manual annotation of the training set. The second is how to calculate the error rate of each individual word aligner. We solve this by calculating the error rate of a manually annotated held-out data set instead of the entire training set. In addition, the final ensemble takes into account the weights of the alignment links produced by the individual word aligners. Experimental results indicate that the boosting method proposed in this paper performs much better than the original word aligner, achieving a large error rate reduction.

1 Introduction

Bilingual word alignment is first introduced as an intermediate result in statistical machine translation (SMT) (Brown et al., 1993). In recent years, some researchers have employed statistical word alignment models to build alignment links (Brown et al., 1993; Wu, 1997; Och and Ney, 2000; Cherry and Lin, 2003). Other researchers used similarity and association measures to build alignment links (Ahrenberg et al., 1998; Tufis and Barbu, 2002).

One issue about word alignment is how to improve the performance of a word aligner when the training data is fixed. One possible solution is to use boosting (Freund and Schapire, 1996), which is one of the ensemble methods (Dietterich, 1997 & 2000). Boosting generates the classifiers sequentially and changes the weight of the training instance that is provided as input to each inducer based on the previously built classifiers.

The underlying idea of boosting is to combine simple “rules” to form an ensemble such that the

performance of the single ensemble is improved. This idea of boosting has its roots in PAC learning. Kearns and Valiant (1994) proved that learners, each performing only slightly better than random, can be combined to form an arbitrarily good ensemble hypothesis. The AdaBoost (**Adaptive Boosting**) algorithm by Freund and Schapire (1996 & 1997) is generally considered as the first step towards practical boosting algorithms. Recently, boosting has been used in many NLP applications. For instance, it has been applied to tagging and PP attachment (Abney et al., 1999), word sense disambiguation (Escudero et al., 2000), parser construction (Haruno et al., 1998; Henderson and Brill, 2000), and sentence generation (Walker et al., 2001).

In this paper, we investigate how to improve word alignment with boosting. Although word alignment is not a classification problem, we can still build different word aligners by changing the weights of the training samples. If these aligners perform accurately and diversely on the corpus (Dietterich, 2000), they can be employed to improve the word alignment results. Applying boosting to word alignment must solve two problems. The first is how to automatically build the reference set for the training data since manually annotating the training set is a heavy labor work. We build it automatically by using bi-directional word alignment results. The second is how to calculate the error rate of each individual word aligner. We solve it by calculating the error rate of a manually annotated held-out data set instead of the entire training set. In addition, besides using the weight of the individual aligner, we also use the weights of the alignment links produced by the word aligner. Experimental results indicate that our method performs much better than the original word aligner, achieving an error rate reduction of 10.28% and 21.52% in the English to Chinese direction and in the Chinese to English direction, respectively.

The remainder of the paper is organized as follows. Section 2 introduces the basic idea of the AdaBoost algorithm. Section 3 describes our boosting method used for word alignment,

including reference set construction, error rate calculation, and the alignment algorithm. Section 4 presents the evaluation results. Section 5 discusses the boosting approach proposed in this paper. The last section concludes this paper and presents the future work.

2 Boosting

The AdaBoost algorithm is well known in machine learning. It imposes one constraint on its underlying learner: it may abstain from making predictions about labels of some samples, but it must consistently be able to get more than 50% accuracy on the samples for which it commits to a decision. Its accuracy is measured according to the distribution describing the importance of samples. The learners must be able to get more correct samples than incorrect samples. It is called the *weak learning criterion*.

The detailed algorithm of AdaBoost is described in Figure 1. At each round, the learner makes predictions on weighted samples. The error rate of the learner is calculated based on the predictions. According to the error rate, the weights are recalculated for the next round. Thus, boosting re-samples the training data by changing the weights of the data.

3 Boosting the Statistical Word Aligner

In this section, we apply the AdaBoost algorithm to improve statistical word alignment. The statistical word aligner is used as the learner.

Applying AdaBoost to word alignment, two issues should be considered. AdaBoost is a supervised learning algorithm. Thus, the first is how to construct the reference set for the large-scale bilingual training corpus. The second is how to calculate the alignment error rate of each word aligner on each round.

In the following sections, we first describe how to construct the reference set for the training data and how to calculate the error rate, and then describe the algorithm to boost a word aligner.

3.1 Reference Set Construction

For statistical word alignment, large-scale training data is needed to obtain good results. However, the AdaBoost algorithm requires that the training data be annotated manually. This needs heavy manual work. In order to tackle with this problem, we build a pseudo reference set for the training data.

Here, we build the reference set by using bi-directional word alignment results of the training data. The statistical word alignment model used is IBM model 4 as described in (Brown, et al., 1993).

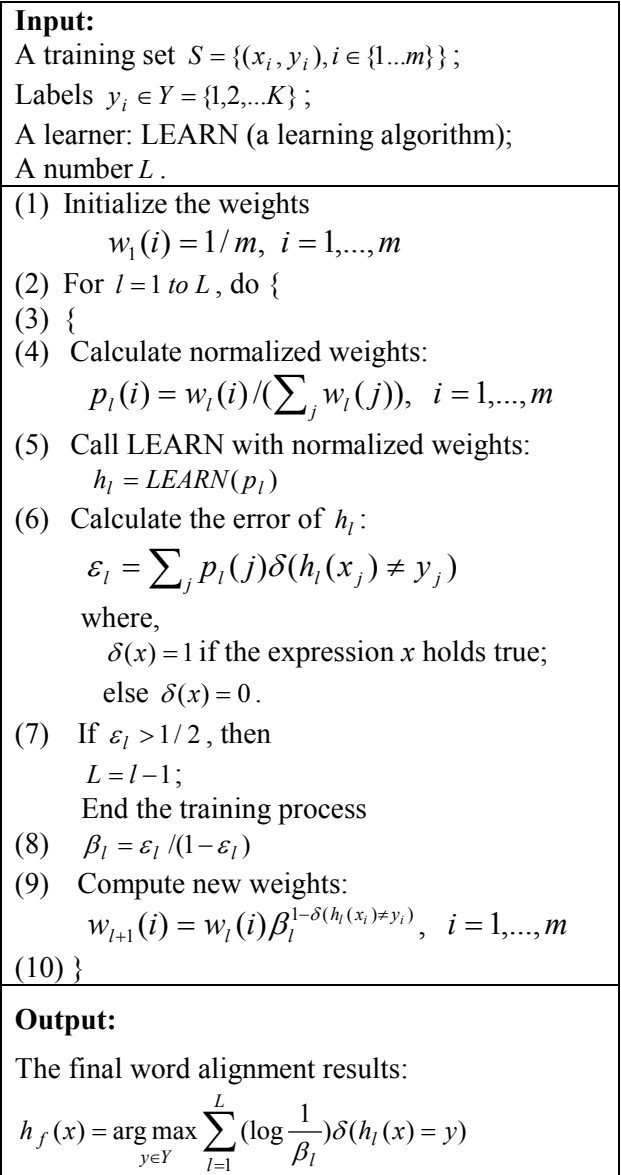


Figure 1. The AdaBoost Algorithm

The GIZA++¹ toolkit is used to perform statistical alignment. Bi-directional word alignment includes alignment in two directions (source to target and target to source) as described in (Och and Ney 2000). Thus, for each sentence pair, we get two alignment results. We use S_1 and S_2 to represent the bi-directional alignment result sets. For alignment links in both sets, we use i for source words and j for target words.

$$S_1 = \{(A_j, j) \mid A_j = \{i \mid i = a_j, a_j \geq 0\}\} \quad (1)$$

$$S_2 = \{(i, A_i) \mid A_i = \{j \mid a_j = i, a_j \geq 0\}\} \quad (2)$$

Where, a_j represents the index position of the source word aligned to the target word in position j .

¹ It is located at <http://www.fjoch.com/GIZA++.html>.

For example, if a target word in position j is aligned to a source word in position i , then $a_j = i$.

If a target word in position j is aligned to source words in positions i_1 and i_2 , then $A_j = \{i_1, i_2\}$. We name an element in the alignment set an *alignment link*.

The intersection of the two alignment sets is considered as reliable, and therefore, directly used as a part of the reference set. It is represented as shown in (3).

$$I = S_1 \cap S_2 \quad (3)$$

The precision of the intersection set is high, but its recall is low (Och and Ney, 2000)². Thus, we consider improving the recall of the reference set. Based on word alignment results in $S_1 \cup S_2$, we calculate the translation probabilities of the aligned word pairs as shown in (4).

$$p(t | s) = \frac{\text{count}(s, t)}{\sum_t \text{count}(s, t)} \quad (4)$$

Where, $\text{count}(s, t)$ is the occurring frequency of the alignment link $(s, t) \in S_1 \cup S_2$.

Those word alignment links whose translation probabilities are above a threshold δ_1 and whose occurring frequencies are above a threshold δ_2 are added to the reference set. This set C is represented as in (5).

$$C = \{(s, t) | p(t | s) > \delta_1 \ \& \ \text{count}(s, t) > \delta_2\} \quad (5)$$

Combining the results in (3) and (5), we obtain the reference set R of the training data as shown in (6).

$$R = I \cup C \quad (6)$$

3.2 Error Rate Calculation

For word alignment, a sentence pair in the training set is a training sample. Thus, we first calculate the error rate of each sentence pair. Their weighted sum is used as the error rate of the word aligner on each round. Och and Ney (2000) defined a method to calculate the alignment error rate (AER). It is modified by Wu and Wang (2004). We use the version described in (Wu and Wang 2004). The error rate is described in (7).

$$AER(i) = 1 - \frac{2 * |S_G \cap R_S|}{|S_G| + |R_S|} \quad (7)$$

Where S_G represents the set of alignment links of a sentence pair i identified by the proposed method. R_S denotes the reference alignment set for the sentence pair.

As described in the above section, we build a pseudo reference set for the training data. Thus, the error rate calculated according to this reference set is not the real one. In order to partly solve this problem, we use a held-out data set, which is randomly selected from the training data. We manually annotate this held-out set instead of the training set to get its real reference set. On each round, we calculate the error rates of the sentence pairs in this held-out set, the weighted sum of which is used as the error rate of each word aligner. Based on the error rate and the pseudo reference set, the weight of each sentence pair can be updated on each round.

3.3 Word Alignment Algorithm

After defining the reference set and the alignment error rate, we describe the algorithm to boost the word aligner in this section. The learner is the IBM model 4. The detailed boosting algorithm is shown in Figure 2.

This algorithm has four different points as compared with that in Figure 1. First, the reference set for the training data is different on each round. We train two alignment results (source to target, and target to source) on each round and get the pseudo reference set. Second, the error rates are calculated according to a held-out set instead of the training data. Third, the weight updating rule is different. In a sentence pair, there are several word alignment links. Some are correct, and others are incorrect. Thus, we update the weights according to the number of correct and incorrect alignment links. The last is about the construction of the final ensemble. Besides using $\log \frac{1}{\beta_t}$ as the confidence

measure of the word aligner on each round, we also use the weight $WT_t(s, t)$ to measure the confidence of each alignment link produced by each word aligner. The weight $WT_t(s, t)$ will be described in the next section.

²On our testing set, the precision is 0.9274 and the recall is 0.5765.

<p>Input:</p> <p>A training set S including m bilingual sentence pairs; A held-out data set D ; The alignment reference set R ; The word aligner (the learner LEARN); A number L.</p>
<p>(1) Initialize the weights: $w_1(i) = 1/m, i = 1, \dots, m$</p> <p>(2) For $l = 1$ to L, do</p> <p>(3) {</p> <p>(4) For each sentence pair i, compute normalized weights on both the training set and held-out set: $p_l(i) = w_l(i) / \sum_j w_l(j), i = 1, \dots, m$</p> <p>(5) Perform word alignment with normalized weights: $h_l = LEARN(p_l)$ Update the pseudo reference set R.</p> <p>(6) Calculate the error of h_l using the held-out set D : $\varepsilon_l = \sum_{j \in D} p_l(j) * AER(j)$ $AER(j)$ is calculated as in Equation (7).</p> <p>(7) If $\varepsilon_l > 1/2$, then $L = l - 1$; End the training process</p> <p>(8) $\beta_l = \varepsilon_l / (1 - \varepsilon_l)$</p> <p>(9) For all i, compute new weights: $w_{l+1}(i) = w_l(i) * (k + (n - k) * \beta_l) / n$ where, n represents n alignment links in the i^{th} sentence pair. k represents the number of error links compared with the pseudo reference set R.</p> <p>(10) }</p>
<p>Output: The final word alignment result for a source word or phrase s:</p> $h_f(s) = \arg \max_t \sum_{l=1}^L (\log \frac{1}{\beta_l}) * WT_l(s, t) * \delta(h_l(s) = t)$ <p>where, $\delta(x) = 1$ if the expression x holds true; else $\delta(x) = 0$. $WT_l(s, t)$ is the weight of the alignment link (s, t) produced by the word aligner h_l.</p>

Figure 2. The Boosting Algorithm for Word Alignment

3.4 Weight Calculation

The algorithm in Figure 2 shows that the weights are related with the specific word

alignment links and the specific word aligner. We calculate the weights $WT_l(s, t)$ based on the word alignment results of the training data.

On each round, we get a different alignment results on the training data. Based on the word alignment results, we calculate the weight of each word alignment link (s, t) produced by the word aligner h_l as shown in (8).

$$WT_l(s, t) = \frac{2 * count(s, t)}{\sum_{t'} count(s, t') + \sum_{s'} count(s', t)} \quad (8)$$

Where, $count(s, t)$ is the occurring frequency of the alignment link (s, t) in the word alignment results of the training data on each round.

In fact, we can also use the alignment probabilities of the alignment model as the voting weights. These alignment probabilities consist of three parts: word translation probability, distortion probability, and fertility probability. However, this weight scheme does not produce better results than that proposed in Equation (8) in our experiments. Thus, in the following section, if we mention the weight of an alignment link, we refer to the weights described in Equation (8) unless explicitly stated.

4 Evaluation

4.1 Training and Testing Set

We perform experiments on a sentence aligned English-Chinese bilingual corpus collected from general domain. There are about 320,000 bilingual sentence pairs in the corpus, from which, we randomly select 1,000 sentence pairs as testing data and select another 1,500 sentence pairs as the held-out data set. The remainder is used as training data. In the sentence pairs, the average length of the English sentences is 13.6 words while the average length of the Chinese sentences is 14.2 words.

The Chinese sentences among the training set, the held-out set, and the testing set are automatically segmented into words. The segmentation errors in the testing set are post-corrected. Both the held-out set and the testing set are manually annotated. There are totally 8,651 alignment links in the testing set. Among them, 866 alignment links include multiword units, which accounts for about 10% of the total links.

4.2 Evaluation metrics

We use the same evaluation metrics as described in (Wu and Wang, 2004). If we use S_G to represent the set of alignment links identified by the proposed methods and S_C to denote the

reference alignment set, the methods to calculate the precision, recall, f-measure, and alignment error rate (AER) are shown in Equation (9), (10), (11), and (12). From the evaluation metrics, it can be seen that the higher the f-measure is, the lower the alignment error rate is. Thus, we will only show precision, recall and AER scores in the experimental results.

$$precision = \frac{|S_G \cap S_C|}{|S_G|} \quad (9)$$

$$recall = \frac{|S_G \cap S_C|}{|S_C|} \quad (10)$$

$$fmeasure = \frac{2*|S_G \cap S_C|}{|S_G| + |S_C|} \quad (11)$$

$$AER = 1 - \frac{2*|S_G \cap S_C|}{|S_G| + |S_C|} = 1 - fmeasure \quad (12)$$

4.3 Evaluation Results for Boosting

During the experiments, when L is larger than ten, the error rate is larger than 0.5. Thus, we obtain $L=10$. The boosting method will be compared with a *baseline* method. The baseline method runs the word aligner only one round. We build two ensembles: one is trained using the training data in the source to target direction, and the other is trained in the target to source direction. The boosting results are shown in Table 1 and Table 2.

Method	Precision	Recall	AER
Boosting	0.7717	0.7604	0.2340
Baseline	0.7435	0.7350	0.2608

Table 1. Boosting Results for English to Chinese Word Alignment

Method	Precision	Recall	AER
Boosting	0.7713	0.7548	0.2370
Baseline	0.7083	0.6880	0.3020

Table 2. Boosting Results for Chinese to English Word Alignment

From the results, it can be seen that the boosting methods improve the word alignment results. The error rates of the boosting methods performed in both directions are almost the same. However, as compared with its own baseline method, the relative error rate reduction of the boosting method performed in the English to Chinese direction is 10.28%. For the method

performed in the Chinese to English direction, its relative error rate reduction is 21.52%.

In order to further analyze the results, we classify the alignment links into two classes: single word alignment links and multiword alignment links. Single word alignment links only include one-to-one alignments. The multiword alignment links include those having multiword units in the source language or/and in the target language. The results are shown in Table 3 and Table 4, respectively.

Method	Precision	Recall	AER
Boosting (EC)	0.8134	0.8548	0.1664
Baseline (EC)	0.8198	0.8219	0.1792
Boosting (CE)	0.8115	0.8482	0.1706
Baseline (CE)	0.8168	0.7586	0.2134

Table 3. Single Word Alignment Results

Method	Precision	Recall	AER
Boosting (EC)	0.2588	0.1445	0.8145
Baseline (EC)	0.1869	0.1676	0.8233
Boosting (CE)	0.2665	0.1445	0.8126
Baseline (CE)	0.1815	0.2266	0.7985

Table 4. Multiword Alignment Results

In Table 3 and Table 4, the symbols in the brackets of the first columns describe the alignment direction. ‘‘EC’’ and ‘‘CE’’ represents English to Chinese and Chinese to English word alignment, respectively.

For the single word alignment, the boosting methods also achieve lower error rates as compared with the baselines. For example, it achieves 20.06% error rate reduction for CE alignments.

For the multiword alignment, the boosting method does not perform better than the baseline method in the CE direction. For the EC direction, the improvement is also marginal.

4.4 Result Analysis

From the results in the above section, it can be seen that the error rate reduction is very different between the English to Chinese and Chinese to English directions. This is caused by the pseudo reference set of the training set. This reference set is built according to the bi-directional word alignment results (English to Chinese, and Chinese to English).

As described in section 3.1, the pseudo reference set is built according to word alignment links of the two baseline systems in the EC and CE

directions. From the baselines, it can be seen that the word aligner performed in the English to Chinese direction achieves lower error rates than that performed in the Chinese to English direction. In order to further analyze the results, we show the numbers of single word alignment (SWA) links and multiword alignment (MWA) links in the pseudo reference set (RS). We also show how many SWA and MWA links in the reference set are obtained from the links produced by the English to Chinese (EC) baseline system and the Chinese to English (CE) baseline system. The results of the first round are shown in Table 5.

	#SWA	#MWA
EC	1,686,861	40,041
CE	1,610,712	184,529
RS	2,045,943	224,570

Table 5. SWA and MWA Statistics

For single word alignment links, the EC baseline system produces about 82.45% alignment links of the pseudo reference set while the CE baseline system produces about 78.73% alignment links of the pseudo reference set. In addition, 1,251,630 (61.18%) SWA links of the reference set come from the intersection of the EC and CE system.

For the multiword alignment links, the EC baseline system produces about 17.83% alignment links of the reference set while the CE baseline system produces about 82.17% alignment links of the reference set.

The single word alignment of the reference set includes more links from the EC system than from the CE system. Using this reference set, the word aligners performed in the CE direction can be boosted more during the learning procedure. Thus, the boosting ensemble performed in the CE direction achieves much error rate reduction than that method performed in the EC direction.

The multiword alignment links of the reference set are mainly from the CE system. However, the multiword alignment results of both the EC system and the CE system are not greatly improved. For the CE direction, the boosting method does not improve the results because 82.17% of the elements in the reference sets come from itself. Another reason is that the precision of the multiword word alignment in the reference set is very low³. For the EC direction, the word aligner can only produce more-to-one multiword alignment links. However, 82.17% of multiword alignments links in the reference set are one-to-

more alignment links. Thus, these one-to-more alignment links are not useful for the word aligner in the EC direction. As a result, the boosting method does not greatly improve the multiword alignment results of the EC system.

5 Discussion

5.1 Pseudo Reference Set

In this paper, we use a pseudo reference set instead of a manually annotated reference set for the training data. Although this pseudo set also includes some errors, the boosting method still improves word alignment results. In order to analyze the accuracy of the reference set, we compare the pseudo reference set of the held-out data with its manually annotated reference set. The result is shown in Table 6.

	Precision	Recall	AER
SWA+MWA	0.7449	0.6880	0.2847
SWA	0.8463	0.7993	0.1806
MWA	0.2290	0.1832	0.7965

Table 6. Result of the Pseudo Reference Set Compared with the Real Reference Set

The results in Table 6 show that the error rate of the pseudo reference set is 0.2847. The error rate of the multiword word alignment links in the pseudo reference set is very high. This also verifies that the boosting method using such a reference set does not improve the MWA results. The boosting method can improve the SWA results because the AER of the SWA in the reference set is comparably low and the reference set can provide some supervision during the learning process.

5.2 Error Rate

As described in section 3.1, we use the error rate of the held-out data set instead of that of the training set on each round. Thus, the confidence of the individual word aligner is also calculated according to the error rate of the held-out data set. This is based on the assumption that the held-out set can represent the features of the training set, including the vocabulary, the phrases, and the sentence structure.

The above assumption is reasonable because the held-out set and training set are from the same domain. The idea is the same as that we use the testing set in our experiments. The testing set only accounts for a small part of the whole corpus. We also use it to test the effectiveness of the algorithm. In fact, this method is efficient, and the boosting

³ This will be further discussed in the next section.

method improves the word alignment in our experiments.

5.3 Weights

After the training process, the final boosting ensemble provides each word aligner a weight. This weight is directly relevant with the error rates provided by the individual word aligner on each round. The smaller the error rate is, the larger the weight is. This weight can be considered as a confidence measure of the individual word aligner.

Besides the above weight obtained from the training process, the individual word aligner also provides a weight for each alignment link produced by it. This kind of weight measures the association strength of the alignment links. The difference between these two weights is that the first weight is a confidence of the individual word aligner and the second weight is a confidence of the individual alignment link produced by the word aligner. Thus, we combine these two weights in our paper. Table 7 compares our method and the boosting method only using the first weight in the CE direction. From the results, it can be seen that our method achieves much improvement, resulting in a relative error rate reduction of 20.12%.

Method	Precision	Recall	AER
Boosting	0.7334	0.6755	0.2967
Our method	0.7713	0.7548	0.2370

Table 7. Comparison Result of the Two Boosting Methods

6 Conclusion and Future Work

This paper uses the boosting method to create ensembles of statistical word aligners. This boosting method automatically builds a pseudo reference set for the training set, which avoids heavy labor annotation work. When calculating the error rate of the individual aligner on each round, we use a manually annotated held-out data set instead of the entire training set. In addition, the final ensemble takes into account the weight of the alignment links produced by the individual word aligner. Experimental results indicate that the boosting method performs better than the original word aligners, achieving an error rate reduction of 10.28% and 21.52% in the English to Chinese direction and in the Chinese to English direction, respectively.

Further analysis indicates that the boosting method can greatly improve the single word alignment results. However, it does not improve the multiword alignment results. This is because of the low accuracy of the multiword alignment links

in the constructed pseudo reference set. Analysis also shows that adding the weights of alignment links produced by the word aligners greatly improves the word alignment results.

In future work, we will investigate how to automatically construct a reference set of the training set with higher accuracy, especially with higher accuracy of the multiword alignment links.

References

- Steven Abney, Robert E. Schapire, Yoram Singer. 1999. Boosting Applied to Tagging and PP Attachment. *In Proc. of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-1999)*, pages 38-45.
- Lars Ahrenberg, Magnus Merkel, and Mikael Andersson. 1998. A Simple Hybrid Aligner for Generating Lexical Correspondences in Parallel Texts. *In Proc. of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (ACL/COLING-1998)*, pages 29-35.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263-311.
- Colin Cherry and Dekang Lin. 2003. A Probability Model to Improve Word Alignment. *In Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, pages 88-95.
- Thomas G. Dietterich. 1997. Machine Learning Research: Four Current Directions. *AI Magazine*, 18 (4): 97-136.
- Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. *In Proc. of the First International Workshop on Multiple Classifier Systems (MCS-2000)*, pages 1-15.
- Gerard Escudero, Lluís Màrquez, German Rigau. 2000. Boosting Applied to Word Sense Disambiguation. *In Proc. of the 12th European Conference on Machine Learning (ECML-2000)*, pages 129-141.
- Yoav Freund and Robert E. Schapire. 1996. Experiments with a New Boosting Algorithm. *In Proc. of the Thirteenth International Conference on Machine Learning (ICML-1996)*, pages 148-156.
- Yoav Freund and Robert E. Schapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.

Journal of Computer and System Sciences, 55(1):119-139.

- Masahiko Haruno, Satoshi Shirai, and Yoshifumi Ooyama. 1998. Using Decision Trees to Construct a Practical Parser. *In Proc. of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (ACL/COLING-1998)*, pages 505-511.
- John C. Henderson and Eric Brill. 2000. Bagging and Bosting a Treebank Parser. *In Proc. of the First Conference on North American Chapter of the Association for Computational Linguistics (NAACL-2000)*, pages 34-41.
- Michael Kearns and Leslie Valiant. 1994. Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. *Journal of the ACM*, 41(1):67-95.
- Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. *In Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, pages 440-447.
- Dan Tufis and Ana Maria Barbu. 2002. Lexical Token Alignment: Experiments, Results and Application. *In Proc. of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, pages 458-465.
- Marilyn A. Walker, Owen Rambow and Monia Rogati. 2001. *SPoT: A Trainable Sentence Planner*. *In Proc. of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*.
- Dekai Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3): 377-403.
- Hua Wu and Haifeng Wang. 2004. *Improving Domain-Specific Word Alignment with a General Bilingual Corpus*. In Robert E. Frederking and Kathryn B. Taylor (Eds.), *Machine Translation: From Real Users to Research: 6th conference of the Association for Machine Translation in the Americas (AMTA-2004)*, pages 262-271.