Improving Statistical Word Alignment with Ensemble Methods

Hua Wu and Haifeng Wang

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 ensemble methods. Two ensemble methods are investigated: bagging and cross-validation committees. On these two methods, both weighted voting and unweighted voting are compared under the word alignment task. In addition, we analyze the effect of different sizes of training sets on the bagging method. Experimental results indicate that both bagging and cross-validation committees improve the word alignment results regardless of weighted voting or unweighted voting. Weighted voting performs consistently better than unweighted voting on different sizes of training sets.

1 Introduction

Bilingual word alignment is first introduced as an intermediate result in statistical machine translation (SMT) [3]. Besides being used in SMT, it is also used in translation lexicon building [9], transfer rule learning [10], example-based machine translation [14], etc. In previous alignment methods, some researchers employed statistical word alignment models to build alignment links [3], [4], [8], [11], [16]. Some researchers used similarity and association measures to build alignment links [1], [15].

One issue about word alignment is how to improve the performance of a word aligner when the training data are fixed. One possible solution is to use ensemble methods [5], [6]. The ensemble methods were proposed to improve the performance of classifiers. An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (weighted or unweighted voting) to classify new examples. Many methods for constructing ensembles have been developed [5]. One kind of methods is to resample the training examples. These methods include bagging [2], cross-validation committees [12] and boosting [7]. The two former methods generate the classifiers in parallel while boosting generates the classifiers sequentially. In addition, boosting changes the weights of the training instance that is provided as input to each inducer based on the previously built classifiers.

In this paper, we propose an approach to improve word alignment with ensemble methods. Although word alignment is not a classification problem, we can still build different word aligners by resampling the training data. If these aligners perform accurately and diversely on the corpus [6], they can be employed to improve the word alignment results. Here, we investigate two ensemble methods: bagging and

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 462 – 473, 2005.

cross-validation committees. For both of the ensemble methods, we employ weighted and unweighted voting to build different ensembles. Experimental results indicate that both bagging and cross-validation committees improve the word alignment results. The weighted ensembles perform much better than the unweighted ensembles according to our word alignment results. In addition, we analyze the effect of different sizes of training data on the bagging algorithm. Experimental results also show that the weighted bagging ensembles perform consistently better than the unweighted bagging ensembles on different sizes of training sets.

The remainder of the paper is organized as follows. Section 2 describes statistical word alignment. Section 3 describes the bagging algorithm. Section 4 describes the cross-validation committees. Section 5 describes how to calculate the weights used for voting. Section 6 presents the evaluation results. Section 7 discusses why the ensemble methods used in this paper are effective for the word alignment task. The last section concludes this paper and presents the future work.

2 Statistical Word Alignment

In this paper, we use the IBM model 4 as our statistical word alignment model [3]. This model only allows word to word and multi-word to word alignments. Thus, some multi-word units cannot be correctly aligned. In order to tackle this problem, we perform word alignment in two directions (source to target and target to source) as described in [11]. In this paper, we call these two aligners *bi-directional* aligners.[1] Thus, for each sentence pair, we can get two alignment results. We use S_1 and S_2 to represent the bi-directional alignment sets. For alignment links in both sets, we use *i* for source words and *j* for target words.

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

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

Where, a_j represents the index position of the source word aligned to the target word in position *j*. For example, if a target word in position *j* is connected to a source word in position *i*, then a_j=*i*. If a target word in position *j* is connected 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*.[2]

3 Bagging

The *bagging* algorithm (derived from *b*ootstrap *agg*regat*ing*) votes classifiers generated by different bootstrap replicates [2]. A bootstrap replicate is generated by uniformly sampling *m* instances from the training set with replacement. In general, *T*

[1] The GIZA++ toolkit is used to perform statistical alignment. It is located at http://www.fjoch.com/GIZA++.html.

[2] Our definition of alignment link is different from that in [11]. In [11], alignment links are classified into possible links and sure links. In our paper, both one-to-one and non one-to-one links are taken as sure links.

bootstrap replicates are built in the sampling process. And T different classifiers are built based on the bootstrap replicates. A final classifier is built from these T sub-classifiers using weighted voting or unweighted voting. The original unweighted bagging algorithm is shown in Figure 1.

Input: a training set $S = \{(y_i, x_i), i \in \{1, ..., m\}\}$
an induction algorithm Ψ

(1) For $j = 1$ to T {
(2) S_j = bootstrap replicate of S by sampling m items from S with replacement
(3) $C_j = \Psi(S_j)$
(4) }
(5) Create a final classifier with majority voting:

$$C^*(x) = \arg\max_{y \in Y} \sum_j \delta(C_j(x), y)$$

Where, $\delta(x, y) = 1$ if $x = y$; else $\delta(x, y) = 0$.

Output: Classifier C^*

Fig. 1. The Unweighted Bagging Algorithm

3.1 Bagging the Statistical Word Aligner

In this section, we apply the technique of bagging to word alignment, the detailed algorithm is shown in Figure 2. In the algorithm, we first resample the training data to train the word aligners. We choose to resample the training set in the same way as the original bagging algorithm. With these different bootstrap replicates, we build the different word aligners. As described in Section 2, we perform word alignment in two directions to improve multiword alignment. Thus, on each bootstrap replicate, we train a word aligner in the source to target direction and another word aligner in the target to source direction, which is described in b) of step (1).

After building the different word aligners, we combine or aggregate the alignments generated by the individual alignment models to create the final alignments for each sentence pair. In this paper, the final alignment link for each word is chosen by performing a majority voting on the alignments provided by each instance of the model. The majority voting can be weighted or unweighted. For weighted voting, the weights of word alignment links produced by the bi-directional word aligners are trained from the training data, which will be further described in section 5. For unweighted voting, the best alignment link for a specific word or unit is voted by more than half of the word aligners in the ensemble. For those words that have no majority choice, the system simply does not align them.

Input: a training set $S = \{(y_i, x_i), i \in \{1...m\}\}$
a word alignment model M

(1) For $j = 1$ to T
  a) S_j = bootstrap replicate of S by sampling m items from S with replacement
  b) Train the bi-directional alignment models M_j^{st} and M_j^{ts} with the bootstrap replicate S_j

(2) For $k = 1$ to N (N is the number of sentence pairs)
For each word s:

  a) For weighted voting

$$M^*(s,k) = \arg\max_t \sum_j W_j(s,t) * (\delta(M_j^{st}(s,k),t) + \delta(M_j^{ts}(s,k),t))$$

  t is the word or phrase in the target sentence;
  $W_j(s,t)$ is the weight for the alignment link (s, t) produced by the aligner M_j^{st} or M_j^{ts};
  $\delta(x, y) = 1$ if $x = y$; else $\delta(x, y) = 0$.

  b) For unweighted voting

$$M^*(s,k) = \arg\max_{t:n(t)>\frac{T}{2}} \sum_{j=1}^{T} (\delta(M_j^{st}(s,k),t) + \delta(M_j^{ts}(s,k),t))$$

where, n(t)= $\sum_{j=1}^{T} (\delta(M_j^{st}(s,k),t) + \delta(M_j^{ts}(s,k),t))$

Output: The final word alignment results

Fig. 2. The Bagging Algorithm for Word Alignment

4 Cross-Validation Committee

The difference between bagging and cross-validation committees lies in the way to resample the training set. The cross-validation committees construct the training sets by leaving out disjoint subsets of the training data. For example, the training set can be randomly and evenly divided into *N* disjoint subsets. Then *N* overlapping training sets can be constructed by dropping out a different one of these *N* subsets. This procedure is the same as the one to construct training sets for N-fold cross-validation. Thus, ensembles constructed in this way are called cross-validation committees.

For word alignment, we also divide the training set into *N* even parts and build *N* overlapping training sets. With the *N* sets, we build *N* alignment models as described

above. Since the training sets are different, the word alignment results may be different for individual words. Using the same majority voting as described in Figure 2, we get the final word alignment results.

5 Weight Calculation

In this paper, we compare both weighted voting and unweighted voting under our word alignment task. 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 based on the word alignment results on the training data.

As described in Section 3.1, on each bootstrap replicate *j*, we train a word aligner M_j^{st} in the source to target direction and a word aligner M_j^{ts} in the target to source direction. That is to say, we obtain two different word alignment sets S_j^{st} and S_j^{ts} for each of the bootstrap replicate. For each word alignment link (s, t) produced by M_j^{st} or M_j^{ts}, we calculate its weight as shown in (3). This weight measures the association of the source part and the target part in an alignment link. This measure is like the Dice Coefficient. Smadja et al. [13] showed that the Dice Coefficient is a good indicator of translation association.

$$W_i(s,t) = \frac{2 * count(s,t)}{\sum_{t'} count(s,t') + \sum_{s'} count(s',t)} \tag{3}$$

Where, $count(s,t)$ is the occurring frequency of the alignment link $(s, t) \in S_j^{st} \cup S_j^{ts}$.

6 Experiments

6.1 Training and Testing Set

We perform experiments on a sentence aligned English-Chinese bilingual corpus in 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. 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 in both the training set and the testing set are automatically segmented into words. The segmentation errors in the testing set are post-corrected. The testing set is manually annotated. It has totally 8,651 alignment links. Among them, 866 alignment links include multiword units, which accounts for about 10% of the total links.

6.2 Evaluation Metrics

We use the same evaluation metrics as in [17]. If we use S_G to represent the set of alignment links identified by the proposed methods and S_R to denote the reference

alignment set, the methods to calculate the precision, recall, f-measure, and alignment error rate (AER) are shown in Equation (4), (5), (6), and (7). In addition, t-test is used for testing statistical significance. 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.

$$precision = \frac{|S_G \cap S_R|}{|S_G|} \tag{4}$$

$$recall = \frac{|S_G \cap S_R|}{|S_R|} \tag{5}$$

$$fmeasure = \frac{2*|S_G \cap S_R|}{|S_G| + |S_R|} \tag{6}$$

$$AER = 1 - \frac{2*|S_G \cap S_R|}{|S_G| + |S_R|} = 1 - fmeasure \tag{7}$$

6.3 Evaluation Results for Bagging

For the bagging method, we use ten word aligners trained on five different bootstrap replicates. Among them, five aligners are trained in the source to target direction. The other five aligners are trained in the target to source direction. The bagging method will be compared with a *baseline* method using the entire training data. For this baseline method, we also train bi-directional models. Based on the alignment results on the entire training data, we calculate the alignment weights for the two word aligners as described in Section 5.

The results using weighted voting are shown in Table 1. The number in brackets of the first column describes the number of word aligners used in the ensembles. For example, in the ensemble "bagging (4)", two word aligners are trained in the source to target direction and the other two are trained in the target to source direction.

From the results, it can be seen that the bagging methods obtain significantly better results than the baseline. The best ensemble achieves an error rate reduction of 7.34% as compared with the baseline. The results show that increasing the number of word aligner does not greatly reduce the word alignment error rate. The reduction is even smaller when the number increases from 8 to 10.

Table 1. Weighted Bagging Results

Method	Precision	Recall	AER
Bagging (4)	0.8035	0.7898	0.2034
Bagging (6)	0.8048	0.7922	0.2015
Bagging (8)	0.8061	0.7948	0.1996
Bagging (10)	0.8064	0.7948	0.1994
Baseline	0.7870	0.7826	0.2152

In order to further analyze the effect of the weights on the word alignment results, we also use unweighted voting in the ensembles. The results are shown in Table 2. The *baseline* method also trains bi-directional aligners with the entire training data. The final word alignment results are obtained by taking an unweighted voting on the two alignment results produced by the bi-directional aligners. That is the same as that by taking the intersection of the two word alignment results.

Table 2. Unweighted Bagging Results

Method	Precision	Recall	AER
Bagging (4)	0.9230	0.6073	0.2674
Bagging (6)	0.9181	0.6200	0.2598
Bagging (8)	0.9167	0.6307	0.2527
Bagging (10)	0.9132	0.6347	0.2511
Baseline	0.9294	0.5756	0.2810

Increasing the number of word aligners in the ensembles, the unweighted bagging method does not greatly reduce AER. However, the ensembles obtain much lower error rate as compared with the baseline. The best ensemble achieves a relative error rate reduction of 10.64%, indicating a significant improvement. From the experimental results, we find that there are no multiword alignment links selected in the ensembles. This is because unweighted voting in this paper requires more than half of the word aligners in the ensembles to vote for the same link. Thus, there should be bi-directional word aligners voting for the target alignment link. The intersection of bi-directional word alignment results produced by the IBM models only creates single word alignments. It can also be seen from the Equations (1) and (2) in Section 2.

Comparing the results obtained using weighted voting in Table 1 and those obtained using unweighted voting in Table 2, we find that (1) the weighted bagging methods are much better than the unweighted bagging methods; (2) the ensembles using unweighted voting obtain higher precision but lower recall than those using weighted voting. For example, the weighted voting "bagging (10)" achieves a relative error rate reduction of 20.59% as compared with the corresponding unweighted voting. This indicates that the method used to calculate voting weights described in section 5 is very effective.

6.4 Evaluation Results for Cross-Validation Committees

For the cross-validation committees, we divide the entire training data into five disjoint subsets. For each bootstrap replicate, we leave one out. Thus, each replicate includes 80% sentence pairs of the full training data. For each replicate, we train bi-directional word alignment models. Thus, we totally obtain ten individual word aligners. The baseline is the same as shown in Table 1. The results obtained using weighted voting are shown in Table 3. The number in the brackets of the first column describes the number of word aligners used in the ensembles.

Table 3. Evaluation Results for Weighted Cross-Validation Committees

Method	Precision	Recall	AER
Validation (4)	0.8059	0.7913	0.2015
Validation (6)	0.8070	0.7928	0.2002
Validation (8)	0.8063	0.7933	0.2002
Validation (10)	0.8068	0.7947	0.1993
Baseline	0.7870	0.7826	0.2152

From the results, it can be seen that the cross-validation committees perform better than the baseline. The best ensemble "validation (10)" achieves an error rate reduction of 7.39% as compared with the baseline, indicating a significant improvement. The results also show that increasing the number of word aligner does not greatly reduce the word alignment error rate.

As described in section 6.3, we also use unweighted voting for the cross-validation committees. The results are shown in Table 4. The baseline is the same as described in Table 2.

Table 4. Evaluation Results for Unweighted Cross-Validation Committees

Method	Precision	Recall	AER
Validation (4)	0.9199	0.5943	0.2779
Validation (6)	0.9174	0.6124	0.2655
Validation (8)	0.9154	0.6196	0.2610
Validation (10)	0.9127	0.6245	0.2584
Baseline	0.9294	0.5756	0.2810

From the results, it can be seen that increasing the number of word aligners in the ensembles, the alignment error rate is reduced. The best ensemble achieves a relative error rate reduction of 8.04% as compared with the baseline, indicating a significant improvement. Comparing the results in Table 3 and Table 4, we find that the weighted methods are also much better than the unweighted ones. For example, the weighted method "Validation (10)" achieves an error rate reduction of 22.87% as compared with the corresponding unweighted method.

6.5 Bagging vs. Cross-Validation Committees

According to the evaluation results, bagging and cross-validation committees achieve comparable results. In order to further compare bagging and cross-validation committees, we classify the alignment links in the weighted ensembles into two classes: single word alignment links (SWA) and multiword alignment links (MWA). SWA links only include one-to-one alignments. MWA links refer to those including multiword units in the source language or/and in the target language. The SWA and MWA for the bagging ensembles are shown in Table 5 and Table 6. The SWA and MWA for the cross-validation committees are shown in Table 7 and Table 8. The AERs of the baselines for SWA and MWA are 0.1531 and 0.8469, respectively.

Table 5. Single Word Alignment Results for the Weighted Bagging Methods

Method	Precision	Recall	AER
Bagging (4)	0.8263	0.8829	0.1463
Bagging (6)	0.8270	0.8845	0.1452
Bagging (8)	0.8270	0.8877	0.1437
Bagging (10)	0.8265	0.8876	0.1440

Table 6. Multiword Alignment Results for the Weighted Bagging Methods

Method	Precision	Recall	AER
Bagging (4)	0.4278	0.1815	0.7451
Bagging (6)	0.4432	0.1896	0.7344
Bagging (8)	0.4540	0.1884	0.7336
Bagging (10)	0.4620	0.1896	0.7311

Table 7. Single Word Alignment Results for Weighted Cross-Validation Committees

Method	Precision	Recall	AER
Validation (4)	0.8282	0.8833	0.1452
Validation (6)	0.8285	0.8847	0.1443
Validation (8)	0.8275	0.8851	0.1447
Validation (10)	0.8277	0.8867	0.1438

Table 8. Multiword Alignment Results for Weighted Cross-Validation Committees

Method	Precision	Recall	AER
Validation (4)	0.4447	0.1908	0.7330
Validation (6)	0.4538	0.1931	0.7291
Validation (8)	0.4578	0.1942	0.7273
Validation (10)	0.4603	0.1942	0.7268

From the results, it can be seen that the single word alignment results are much better than the multiword alignment results for both of the two methods. This indicates that it is more difficult to align the multiword units than to align single words.

Comparing the bagging methods and validation committees, we find that these two methods obtain comparable results on both the single word alignment links and multiword alignment links. This indicates that the different resampling methods in these two ensemble methods do not much affect the results on our word alignment task.

6.6 Different Sizes of Training Data

In this section, we investigate the effect of the size of training data on the ensemble methods. Since the difference between bagging and cross-validation committees is very small, we only investigate the effect on the bagging ensembles.

We randomly select training data from the original training set described in Section 6.1 to construct different training sets. We construct three training sets, which include 1/4, 1/2 and 3/4 of sentence pairs of the original training set, respectively.

For each of the training set, we obtain five bootstrap replicates and train ten word aligners. The results of ensembles consisting of ten word aligners are shown in Table 9 and Table 10. Table 9 and Table 10 show the weighted and unweighted bagging results, respectively. The methods to construct the baselines for different training sets in Table 9 and Table 10 are the same as those in Table 1 and Table 2, respectively. For convenience, we also list the results using the original training set in the tables. The first column describes the size of the training sets used for the ensembles. The last column presents the relative error rate reduction (RERR) of the ensembles as compared with the corresponding baselines. From the results, it can be seen that both weighted and unweighted bagging ensembles are effective to improve word alignment results. The weighted ensembles perform consistently better than the unweigted ensembles on different sizes of training sets.

Table 9. Weighted Bagging Results on Different Sizes of Training Sets

Data	Precision	Recall	AER	Baseline (AER)	RERR
1/4	0.7684	0.7517	0.2316	0.2464	6.00%
1/2	0.7977	0.7775	0.2125	0.2293	7.33%
3/4	0.8023	0.7869	0.2055	0.2184	5.89%
All	0.8064	0.7948	0.1994	0.2152	7.34%

Table 10. Unweighted Bagging Results on Different Sizes of Training Sets

Data	Precision	Recall	AER	Baseline (AER)	RERR
1/4	0.8960	0.6033	0.2789	0.3310	15.72%
1/2	0.9077	0.6158	0.2662	0.3050	12.72%
3/4	0.9140	0.6270	0.2562	0.2943	12.95%
All	0.9132	0.6347	0.2511	0.2810	10.64%

7 Discussion

Both bagging and cross-validation committees utilize multiple classifiers to make different assumptions about the learning system. Bagging requires that the learning system should not be stable, so that small changes to the training set would lead to different classifiers. Breiman [2] also noted that poor predicators could be transformed into worse ones by bagging.

In this paper, the learning system is the word alignment model described in Section 2. The classifiers refer to the different word aligners trained on different bootstrap replicates. In our experiments, although word alignment models do not belong to unstable learning systems, bagging obtains better results on all of the datasets. This is

because the training data is insufficient or subject to data sparseness problem. Thus, changing the training data or resampling the training data causes the alternation of the trained parameters of the alignment model. The word aligners trained on a different bootstrap replicate produce different word alignment links for individual words. Using majority voting, the ensembles can improve the alignment precision and recall, resulting in lower alignment error rates.

The experiments also show that weighted voting is better than unweighted voting. The advantage of weighted voting is that it can select the good word alignment link even if only one aligner votes for it in the ensembles. This is because the selected alignment link gets much higher weight than the other links.

8 Conclusion and Future Work

Two ensemble methods are employed in this paper to improve word alignment results: bagging and cross-validation committees. Both of these two methods obtain better results than the original word aligner without increasing any training data. In this paper, we use two different voting methods: weighted voting and unweighted voting. Experimental results show that the weighted bagging method and weighted cross-validation committees achieve an error rate reduction of 7.34% and 7.39% respectively, as compared with the original word aligner. Results also show that weighted voting is much better than unweighted voting on the word alignment task. Unweighted voting obtains higher precision but lower recall than weighted voting. In addition, the weighted voting used in this paper obtains multiword alignment links while the unweighted voting cannot.

We also compare the two ensemble methods on the same training data and testing data. Bagging and cross-validation committees obtain comparable results on both single word alignment links and multiword alignment links. This indicates that the different resampling methods in these two ensemble methods do not much affect the results under our word alignment task.

We also investigate the bagging method on different sizes of training sets. The results show that both weighted voting and unweighted voting are effective to improve word alignment results. Weighted voting performs consistently better than unweigted voting on different sizes of training sets.

In future work, we will investigate more ensemble methods on the word alignment task such as the boosting algorithm. In addition, we will do more research on the weighting schemes in voting.

References

1. Ahrenberg, L., Merkel, M., Andersson, M.: 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 Int. Conf. on Computational Linguistics (ACL/COLING-1998), 29-35
2. Breiman, L.: Bagging Predicators. Machine Learning (1996), 24(1): 123-140
3. Brown, P. F., Pietra, S. D., Pietra, V. D., Mercer, R.: The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics (1993), 19(2): 263-311

4. Cherry, C., Lin, D.: A Probability Model to Improve Word Alignment. In Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003), pp. 88-95
5. Dietterich, T.: Machine Learning Research: Four Current Directions. AI Magazine (1997), 18 (4): 97-136
6. Dietterich, T.: Ensemble Methods in Machine Learning. In Proc. of the First Int. Workshop on Multiple Classifier Systems (2000), 1-15
7. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In Machine Learning: Proc. of the Thirteenth International Conference (1996), 148-156
8. Matusov, E., Zens, R., Ney H.: Symmetric Word Alignments for Statistical Machine Translation. In Proc. of the 20th Int. Conf. on Computational Linguistics (COLING-2004), 219-225
9. Melamed, I. D.: Automatic Construction of Clean Broad-Coverage Translation Lexicons. In Proc. of the 2nd Conf. of the Association for Machine Translation in the Americas (AMTA-1996), 125-134
10. Menezes, A., Richardson, S.D.: A Best-first Alignment Algorithm for Automatic Extraction of Transfer Mappings from Bilingual Corpora. In Proc. of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation (2001), 39-46
11. Och, F. J., Ney, H.: Improved Statistical Alignment Models. In Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000), 440-447
12. Parmanto, B., Munro, P., Doyle, H.: Improving Committee Diagnosis with Resampling Techniques. In Touretzky, D., Mozer, M., Hasselmo, M. (Ed..): Advances in Neural Information Processing Systems (1996), Vol. 8, 882-888
13. Smadja, F. A., McKeown, K. R., Hatzivassiloglou, V.: Translating Collocations for Bilingual Lexicons: a Statistical Approach. Computational Linguistics (1996), 22 (1):1-38
14. Somers, H.: Review Article: Example-Based Machine Translation. Machine Translation (1999), 14: 113-157
15. Tufis, D., Barbu, M.: Lexical Token Alignment: Experiments, Results and Application. In Proc. of the 3rd Int. Conf. on Language Resources and Evaluation (LREC-2002), 458-465
16. Wu, D.: Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. Computational Linguistics (1997), 23(3): 377-403
17. Wu, H., Wang, H.: Improving Domain-Specific Word Alignment with a General Bilingual Corpus. In Frederking R., Taylor, K. (Eds.): Machine Translation: From Real Users to Research: 6th Conf. of the Association for Machine Translation in the Americas (AMTA-2004), 262-271