PAPER

An Empirical Study of Classifier Combination Based Word Sense Disambiguation

Wenpeng LU[†a)], ***Nonmember***, **Hao WU**[††b)], ***Member***, **Ping JIAN**[††c)], **Yonggang HUANG**[††d)], ***and*** **Heyan HUANG**[††e)], ***Nonmembers***

SUMMARY Word sense disambiguation (WSD) is to identify the right sense of ambiguous words via mining their context information. Previous studies show that classifier combination is an effective approach to enhance the performance of WSD. In this paper, we systematically review state-of-the-art methods for classifier combination based WSD, including probability-based and voting-based approaches. Furthermore, a new classifier combination based WSD, namely the probability weighted voting method with dynamic self-adaptation, is proposed in this paper. Compared with existing approaches, the new method can take into consideration both the differences of classifiers and ambiguous instances. Exhaustive experiments are performed on a real-world dataset, the results show the superiority of our method over state-of-the-art methods.

key words: *word sense disambiguation, classifier combination, probability weighted voting method, self-adaptation*

1. Introduction

There are many ambiguous words in human natural language, which can be interpreted with multiple senses depending on their contexts. Word sense disambiguation (WSD) is to automatically infer the right sense of ambiguous word based on its context. WSD is one of basic tasks in natural language processing (NLP), which is crucial for most NLP applications, such as machine translation, information retrieval, information extraction, content analysis, etc [1]–[3].

From the viewpoint of machine learning, WSD is a typical classification problem. To design a WSD model is to design a classifier. Therefore, WSD can benefit from machine learning community. As observed in studies of classifiers, different models utilize different classification features and algorithms, which causes that the set of patterns misclassified by different classifiers would not necessarily overlap [4]. This means that different classifiers potentially offer complementary information each other. This observation highly motivates the interest in combining multiple classifiers to build an ensemble classifier, which would achieve better performance than individual classifier. Classifier combination is one of main directions in machine learning research [5]–[7], which has received more and more attention and has been widely applied in WSD [8]–[15]. In international workshop on semantic evaluation (SemEval), ensemble classifiers have achieved significantly better performance than individual classifiers [16], [17].

In recent years, the typical works on classifier combination can be divided into probability-based method and voting-based method. The former, such as product rule and sum rule [4], [14], does multiplication or addition operations on the probability of each sense outputted by individual classifier to determine the final probability of each sense; then, the sense with maximum probability is selected as right sense. The latter, such as majority-voting and probability weighted voting method [8]–[10], [12], [13], combines the probability of each sense outputted by individual classifier with different weighted accumulative addition algorithms; then, the sense with maximum accumulative vote is selected as right sense.

Both of probability-based method and voting-based method consider combination strategy from the viewpoint of classifiers. Once combination strategy is selected, it would process all of ambiguous instances with the same combination algorithm and would not adjust any combination parameter for different instances. This ignores the difference between ambiguous instances. Obviously, though a classifier can achieve better performance for the instances of Class *A* and is assigned with a higher weight on this class, it may fail to achieve the same performance on the instances of Class *B*. For different kinds of ambiguous instances, the weight of each classifier and combination strategy should be adjusted. In other words, when multiple classifiers are combined, the influence of difference between ambiguous instances should be considered. Aiming at the problem, Zhang et al. have proposed a weighted voting method with instance dynamic self-adaptation [13]. For each ambiguous instance, the self-confidence of each classifier is evaluated based on the probabilities of senses; based on its self-confidence, the weight of a classifier on each instance would be dynamically adjusted. Zhang's method considers combination strategy from the viewpoint of ambiguous instance, in which, the weight parameter is dynamically assigned, according to self-confidence of the classifier on an ambiguous instance.

Manuscript received March 13, 2017.
Manuscript revised June 16, 2017.
Manuscript publicized August 23, 2017.

[†]The author is with College of Information, Qilu University of Technology (Shandong Academy of Sciences), JiNan, 250353, China.

[††]The authors are with the Department of Computer Science and Technology, Beijing Institute of Technology, Beijing, 100081, China.

a) E-mail: lwp@qlu.edu.cn
b) E-mail: wuhao123@bit.edu.cn
c) E-mail: pjian@bit.edu.cn
d) E-mail: yghuang@bit.edu.cn
e) E-mail: hhy63@bit.edu.cn

DOI: 10.1587/transinf.2017EDP7090

However, there is another deficiency in the method, that is, it completely ignores the difference between overall performances of classifiers. If a classifier's overall performance is poor, even though it shows higher self-confidence on an ambiguous instance, we still should not assign a higher weight to it.

After comparing the differences of existing combination methods, aiming at their deficiencies, the paper presents a novel method to combine classifiers for WSD, that is, the probability weighted voting method with dynamic self-adaptation. Our method takes into account both of the differences between overall performances of classifiers and the differences between ambiguous instances. On coarse-grained English all-words task in SemEval [18], recall of the presented method achieves 83.08%, which is the best performance in all of classifier combination methods.

The contributions of this paper are twofold: a detailed empirical comparison of classifier combination based WSD is reviewed and implemented on the same standard evaluation dataset; furthermore, a novel classifier combination based WSD, that is probability weighted voting method with dynamic self-adaptation, is put forward. The rest of the paper is structured as follows. Section 2 introduces some related works on classifier combination. Section 3 compares the existing state-of-the-art combination methods. Aiming at the deficiency of existing methods, the Probability Weighted Voting method with Dynamic self-Adaptation (PWVDA) is proposed in Sect. 4. The detailed comparison experiments are shown in Sect. 5, which are carried out on fine-grained and coarse-grained levels respectively. At last, the conclusion and future work are described.

2. Related Works

With the development of ensemble learning in the field of machine learning, classifier combination has been widely applied in WSD. Related works about classifier combination are introduced in this section.

According to the similarity and difference between base classifiers, the strategies for classifier combination can be divided into homogeneous and heterogeneous combination. Homogeneous combination utilizes the same kind of base classifiers with different parameters or features to disambiguate instances many times and combines their results. Quan et al. presented a WSD method based on multi-classifier combination with AdaBoost [15]. Firstly, multiple Bayesian classifiers were built, which were trained and updated by labeled and unlabeled examples; then, the sense of ambiguous words were chosen by combining the decisions of classifiers. Wang et al. presented a classifier combination method based on trajectory [19]. Firstly, a series of Bayesian classifiers were constructed with different context windows, which performed sense selection for training and test instances; thus, a sense selection trajectory were created along the sequence of context windows for each instance; then, the trajectories were utilized to make final sense selection with KNN method. Pedersen et al. presented a simple approach to build ensemble classifier with multiple Bayesian classifiers for WSD [20]. The methods of Quan [15], Wang [19] and Pedersen [20] utilize the same kind of base classifiers to construct ensemble classifier, which belong to homogeneous combination strategy. In contrast, heterogeneous combination utilizes different kinds of base classifiers to build ensemble classifier. Florian et al. selected Bayesian, cosine, decision list, transformation-based learning classifiers as base classifiers, then combined them with probability-based voting, confidence-based combination, performance-based combination [8]. Wu et al. utilized product rule and sum rule to combine multiple classifiers, such as support vector machine, naive Bayes, decision trees [14]. According to related literatures [8]–[10], [13], [14], [17], heterogeneous combination is more prevalent than homogeneous combination in the field of classifier combination.

According to the difference between data objects processed by combination methods, the strategies for classifier combination can be divided into probability-based and voting-based combination. Probability-based combination originates from Kittler's work [4]. Kittler et al. developed a common theoretical framework for classifier combination, which derived two basic schemes: product rule and sum rule. Based on the basic schemes, max rule, min rule and average rule were proposed [4]. Wu et al. applied Kittler's theoretical framework to WSD on Chinese, who made a systematic comparison on nine kinds of combination strategies [14]. Le et al. further proposed the combination method based on Dempster's rule and weighted averaging operators [21]. Voting-based combination originates from Kilgarriff's work [17]. Kilgarriff et al. utilized voting schemes to combine the participating systems in SensEval-1, which achieved 66.2% error reduction. Florian et al. proposed enhanced count-based voting, which combined count-based voting and probability mixture model; besides, they proposed confidence-based and performance-based combination [8]. Zhang et al. proposed weighted voting scheme with instance dynamic self-adaptation, which considered the performance of each classifier on special instance to assign its individual weight [13].

Most of the existing works consider combination strategy from the viewpoint of classifiers, which try to assign a uniform weight for each classifier to process all of ambiguous instances. This neglects the performance difference of each classifier on various instances [22]. Zhang et al. considered combination strategy from the viewpoint of instances, which assigned different weight to each classifier, according to its probability on each instance [13]. This neglects the difference of overall performance of each classifier.

In this paper, we review the existing classifier combination methods on WSD. Obviously, none of the existing works considers combination strategy from the viewpoint of both of classifiers and instances at the same time. There is still much room to improve the performance of classifier combination. In order to overcome the drawback, we propose classifier combination WSD based on probabil-

ity weighted voting method with dynamic self-adaptation. The detailed comparison shows that our method can surpass state-of-the-art combination methods.

3. Classical Classifier Combination Methods

In order to compare with all kinds of combination methods, we firstly review the state-of-the-art methods for cclassifier combination in this section.

For sake of illustration, the following symbols are defined. n senses of the ambiguous word w are denoted as $s_1, s_2, \cdots, s_n$; m base classifiers are denoted as $c_1, c_2, \cdots, c_m$; the prior probability of s_i is denoted as $P(s_i)$; the posteriori probability of s_i given by c_j is denoted as $P(s_i|c_j)$.

3.1 Probability-Based Methods

Kittler et al. have proposed the common theoretical framework based on probability, which combines classifiers by sum rule or product rule [4]. Based on Bayes theory, the right sense of ambiguous word w is judged with Eq. (1), which can be rewritten as Eq. (2).

$$\hat{s} = \arg\max_{s_i} P(s_i|c_1, c_2, \ldots, c_m) \tag{1}$$

$$\hat{s} = \arg\max_{s_i} \frac{P(c_1, c_2, \ldots, c_m|s_i)P(s_i)}{\sum_{j=1}^{n} P(c_1, c_2, \ldots, c_m|s_j)P(s_j)} \tag{2}$$

3.1.1 Product Rule (PR)

Assuming that the classifiers are conditionally independent, we can obtain the product rule from Eq. (2), which is represented as Eq. (3).

$$\hat{s} = \arg\max_{s_i} P^{-(m-1)}(s_i) \prod_{j=1}^{m} P(s_i|c_j) \tag{3}$$

3.1.2 Sum Rule

Assuming that the posterior probability $P(s_i|c_j)$ is similar with the prior probability $P(s_i)$, that is, $P(s_i|c_j) = P(s_i)(1 + \delta_{ij})$ and $\delta_{ij} \ll 1$, we can obtain the sum rule by substituting $P(s_i|c_j)$ into product rule, as Eq. (4).

$$\hat{s} = \arg\max_{s_i} \left[(1-m)P(s_i) + \sum_{j=1}^{m} P(s_i|c_j) \right] \tag{4}$$

The sum rule can be approximated with the following inequality, that is, $\min_{j=1}^{m} P(s_i|c_j) \leq \frac{1}{m}\sum_{j=1}^{m} P(s_i|c_j) \leq \max_{j=1}^{m} P(s_i|c_j)$. According to the inequality, we can obtain max, min and average rules from sum rule.

(1) Max Rule (Max)

$$\hat{s} = \arg\max_{s_i} \left[\max_{j=1}^{m} P(s_i|c_j) \right] \tag{5}$$

(2) Min Rule (Min)

$$\hat{s} = \arg\max_{s_i} \left[\min_{j=1}^{m} P(s_i|c_j) \right] \tag{6}$$

(3) Average Rule (Average)

$$\hat{s} = \arg\max_{s_i} \left[\frac{1}{m} \sum_{j=1}^{m} P(s_i|c_j) \right] \tag{7}$$

3.2 Voting-Based Methods

3.2.1 Majority Voting (MV)

Majority voting method is the simplest method to combine multiple classifiers. According to the probability of each sense outputted by base classifier, the sense with highest probability will receive a vote; then, the total votes of each sense are counted, the sense with maximum votes is selected as right sense. The method can be represented with Eq. (8).

$$\hat{s} = \arg\max_{s_i} \sum_{j=1}^{m} \Delta_{(s_i,c_j)} \tag{8}$$

In which, $\Delta_{(s_i,c_j)}$ is the vote value of the sense s_i given by the classifier c_j, which can be obtained with Eq. (9).

$$\Delta_{(s_i,c_j)} = \begin{cases} 1, & \text{if } P(s_i|c_j) = \max_{k=1}^{n} P(s_k|c_j); \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

3.2.2 Rank-Based Voting (RBV)

Majority voting method only utilizes vote result of each classifier, which neglects the order information of sense probability. Rank-based voting method sorts the probabilities of senses on descending order. Each sense will receive a vote value, which is inversely proportional to its order. Then, the total votes of each sense are added up and the sense with maximum votes is selected as right sense. Rank-based voting method can be represented as Eq. (10) and Eq. (11).

$$\hat{s} = \arg\max_{s_i} \sum_{j=1}^{m} \Delta_{(s_i,c_j)} \tag{10}$$

$$\Delta_{(s_i,c_j)} = \frac{1}{rank_{(s_i,c_j)}} \tag{11}$$

In which, $rank_{(s_i,c_j)}$ is the descending order of the sense s_i given by the classifier c_j.

3.2.3 Result Weighted Voting (RWV)

Majority and rank-based voting methods accumulate the votes of each classifier, which neglect their performance differences. The vote weight of each classifier is thought as same. This is not consistent with real situation. Aiming at the problem, result weighted voting method introduces

weight regulating parameter into majority voting method, which is shown as Eq. (12).

$$\hat{s} = \arg\max_{s_i} \sum_{j=1}^{m} (\Delta_{(s_i,c_j)} \times \mu_j) \tag{12}$$

In which,the definition of $\Delta_{(s_i,c_j)}$ is same with Eq. (9), μ_j is weight regulating parameter of the classifer c_j.

3.2.4 Probability Weighted Voting (PWV)

Majority voting, result weighted voting and rank-based voting methods utilize the simplified binary vote result or the sorting order as data objects to combine multiple classifiers. In contrast, probability weighted voting method directly utilizes the posterior probability of each sense given by base classifier as data object and considers the performance weight of each classifier, which can be represented as Eq. (13).

$$\hat{s} = \arg\max_{s_i} \sum_{j=1}^{m} (P(s_i|c_j) \times \mu_j) \tag{13}$$

In which, μ_j is weight regulating parameter of the classifier c_j.

3.3 Weighted Voting with Instance Dynamic Self-Adaptation (WVIDA)

Probability-based and voting-based methods consider combination strategy from the viewpoint of classifiers. All of ambiguous instances would be processed with same weight parameter of the classifier. They neglects performance difference of a classifier on a specific instance.

Aiming at the deficiency, Zhang et al. have proposed weighted voting with instance dynamic self-adaptation [13]. The method can be represented as Eq. (14).

$$\hat{s} = \arg\max_{s_i} \sum_{j=1}^{m} (\beta_j \times P(s_i|c_j)) \tag{14}$$

In which,

$$\beta_j = \begin{cases} 0.7, & \text{if } P(s_i|c_j) \geq \lambda_j; \\ 0.3, & \text{otherwise.} \end{cases} \tag{15}$$

$$\lambda_j = \frac{1}{n} \tag{16}$$

Equation (16) is the regulating threshold, which can be seen as the average of posterior probability of each sense given by the classifier c_j. Equation (15) explains that the weighted factor β_j of the classifier c_j on the sense s_i is set based on the self-confidence of c_j. If $P(s_i|c_j)$ is not smaller than λ_j, this means that the classifier c_j has more confidence to select s_i as right sense, therefore, the weighted factor β_j would be set as larger value 0.7; otherwise, c_j is not confident in s_i, therefore, β_j would be set as smaller value 0.3.

4. Probability Weighted Voting with Dynamic Self-Adaptation (PWVDA)

Both probability-based and voting-based methods consider combination strategy from the viewpoint of classifiers. Once combination strategy is selected, it would process all of ambiguous instances with same algorithm and weight parameters of classifier. This fails to reflect the difference between ambiguous instances. For different instances, the weight of each classifier and combination strategy should be adjusted. That is to say, in multi-classifier combination, the influence of difference between ambiguous instances should be considered.

Though WVIDA has been proposed to solve the problem, it considers combination strategy from the viewpoint of ambiguous instance. WVIDA dynamically adjusts weighted factor according to the self-confidence of a classifier on each sense of ambiguous instance. This fully considers individual difference of a classifier on a specific instance. However, this neglects the difference of overall performances of classifiers. That is to say, if the overall performance of a classifier is worse, though the classifier shows higher self-confidence on a sense, its weighted factor on the sense should not be set as a larger value.

For a perfect combination method, both the overall differences of classifier performances and the individual differences of ambiguous instances should be considered. For this purpose, we propose Probability Weighted Voting method with Dynamic self-Adaptation (PWVDA), which combines the advantages of probability weighted voting method and weighted voting method with instance dynamic self-adaptation. This can give consideration to both overall difference of classifier performance and individual difference of ambiguous instance. The method can be represented with Eq. (17).

$$\hat{s} = \arg\max_{s_i} \sum_{j=1}^{m} (\mu_j \times \beta_j \times P(s_i|c_j)) \tag{17}$$

In which,

$$\beta_j = \begin{cases} \alpha, & \text{if } P(s_i|c_j) \geq (1+\delta)\lambda_j; \\ 0.5, & \text{if } (1-\delta)\lambda_j \leq P(s_i|c_j) < (1+\delta)\lambda_j; \\ 1-\alpha, & \text{if } P(s_i|c_j) < (1-\delta)\lambda_j. \end{cases} \tag{18}$$

In Eq. (17), μ_j and β_j are the regulating parameters of overall performance of base classifier and individual self-confidence on specific ambiguous instance. In Eq. (18), α and δ are regulating parameters of self-confidence of individual classifier on current ambiguous instance; the definition of λ_j is same with Eq. (16).

5. Experiments

5.1 Dataset and Evaluation Criteria

The evaluation dataset is Task#07 (Coarse-grained English All-Words Task) in SemEval-2007, which is a popular and common dataset in WSD field. The dataset consists of 5377 words of running text from five different documents. There are 2269 target ambiguous words in the dataset, which includes 1108 nouns, 591 verbs, 362 adjectives and 208 adverbs. The average polysemy of the dataset with coarse-grained sense inventory is 3.06 and its pairwise agreement is 93.80% [18].

In order to compare with the related works [18], [23]–[26], recall is selected as evaluation criteria. In WSD field, recall is defined as follows [1]: If A is the total number of instances that need to be disambiguated, B is the number of instances that are disambiguated correctly, then the recall R can be computed with $R = B/A$. In our experiment, each classifier is required to assign a sense for every instance. This means that coverage is 100%, recall is equal with precision and F_1, so we only use recall as evaluation criteria to compare different classifiers.

5.2 Experiments on Base Classifier

In previous works, we have developed three WSD methods, which are similarity-based WSD with syntactic parsing [27], WSD based on dependency fitness [28], graph-based WSD with domain knowledge [29]. Based on the methods, three base classifiers (BaseClassifier 1, 2 and 3) are built. Their experimental effectiveness and analysis are introduced in this section.

BaseClassifier1 selects right sense based on its similarity with feature words, which are selected with syntactic parsing [27]. BaseClassifier2 selects right sense according to its dependency fitness, which is measured based on the fitness of sense representative words on dependency constraint set [28]. BaseClassifier3 selects right sense based on knowledge graph [29]. The effectiveness comparison of the three base classifiers is shown in Fig. 1.

As is shown in Fig. 1, from the column of "All", we can compare the overall performances of three classifiers. BaseClassifier3 is the best, BaseClassifier1 is in the second place and BaseClassifier2 is worse. The recalls of them are better than 70%. The performance of them on each document and POS are different greatly. Besides, their principles are different in nature. These mean that they can meet the basic requirements of base classifier in multi-classifier combination [13], [14].

5.3 Experiments on Combination Method and Its Discussion

With different methods in Sects. 3 and 4, the base classifiers are combined respectively. The effectiveness of combination methods are compared in detail in this section.

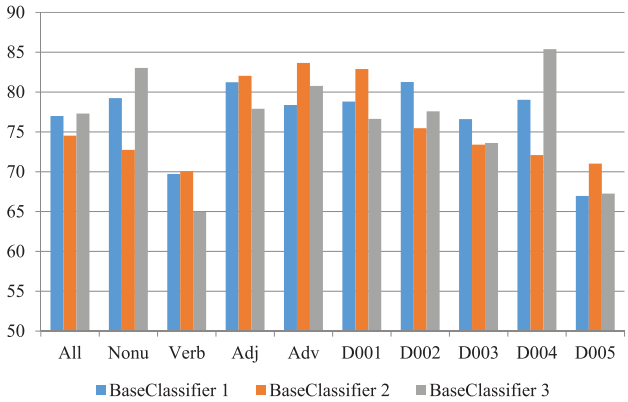

Fig. 1 Effectiveness comparison of base classifiers (recall, %).

5.3.1 Parameters Setting

As is introduced in Sects. 3 and 4, RWV, WV, WVIDA and PWVDA need to set a series of weight parameters. Genetic algorithm is utilized to optimize the parameters. Genetic algorithm is a search heuristic algorithm that mimics the process of natural selection, which generates solutions to optimization problems with techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover. In the experiments, $f(\alpha) = 1 - Recall$ is set as fitness function, whose input variables are set as the weight parameters that need to be optimized.

5.3.2 Effectiveness Comparison on Coarse-Grained and Fine-Grained Senses

The senses in WordNet are fine-grained, whose fine granularity of sense inventories is thought as one of the major obstacles to high-performance WSD [18]. Besides fine-grained senses, the dataset of Task#07 provides the coarse-grained sense inventories. Therefore, the combination experiments can be done on coarse-grained and fine-grained levels respectively. Obviously, the difference of sense levels can directly affect the result of combination classifiers. In the paper, the combination experiments are done on fine-grained and coarse-grained levels respectively, their overall performances are compared in Fig. 2.

As is shown in Fig. 2, the following conclusions can be drawn.

- When the performances of base classifiers and combination methods are compared, it is clear that all combination methods are superior to base classifiers, except that Max method with fine-grained sense is inferior to BaseClassifier1 and BaseClassifier3. This means that it is feasible to improve the effectiveness of WSD with classifier combination methods. Besides, the incogitant combination method may hurt the performance of WSD.

- Except for majority voting (MV), the performance of each combination method on coarse-grained level is superior to that on fine-grained level. The over-fine granularity has been criticized by researchers [18], [23]. To merge the senses moderately is helpful to improve WSD. The combination on coarse-grained level not only takes the coarse-grained sense as the criterion of result evaluation, but also takes the posterior probability of coarse-grained sense as the basis of combination. For combination methods, these provide favorable conditions to obtain better performance on coarse-grained sense level.
- For the probability-based methods (PR, Max, Min, Average), their performances on fine-grained and coarse-grained sense levels are consistent, that is, Average method is best, PR follows, Min and Max are worst.
- For the voting-based methods (MV, RBV, RWV, PWV), on fine-grained sense level, MV and RWV are superior to PWV, RBV is worse. However, on coarse-grained sense level, the performances of MV, RBV, RWV and PWV are improved in turn. MV utilizes the vote value of each classifier, RBV utilizes the order information of sense posterior probability, RWV utilizes the vote value and weight of each classifier, PWV utilizes the posterior probability of each sense and the weight of each classifier. Theoretically, the more detailed posterior probability of each sense given by base classifier is utilized, the better performance of WSD should be achieved. From the viewpoint of coarse-grained sense level, the performances of four methods are improved one by one, which are consistent with the theoretical analysis. However, their performances fail to keep the consistence with theoretical analysis on fine-grained sense level. This may be caused by the negative effect of over-fine granularity of sense inventory, which makes RBV and PWV fail to transfer their votes to the right sense.
- For PWV, WVIDA and PWVDA, PWVDA proposed in the paper is the best and most stable method, which achieves the best performance on fine-grained sense level as well as on coarse-grained sense level. PWV and WVIDA respectively achieve the second performance on fine-grained and coarse-grained levels. PWV considers the combination of multiple classifiers from the viewpoint of classifiers, WVIDA considers the combination from the viewpoint of ambiguous instance. However, PWVDA combines the advantages of PWV and WVIDA, which considers both of the difference of classifiers' overall performance and the difference of ambiguous instances. The comprehensive consideration gives PWVDA the chance to beat PWV and WVIDA.
- Among all of combination methods, no matter whether on fine-grained or coarse-grained sense level, it is apparently that PWVDA proposed in the paper has achieved the best effectiveness. This outstanding performance demonstrates that it is right to improve multi-classifier combination from both the viewpoint of the difference between classifiers' overall performance and that of the difference between ambiguous instances.

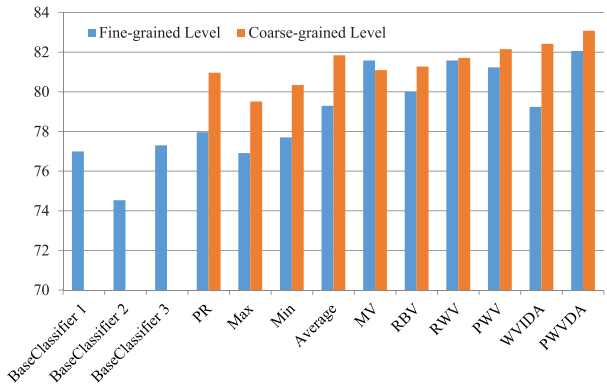

Fig. 2 Effectiveness comparison of all methods (recall, %).

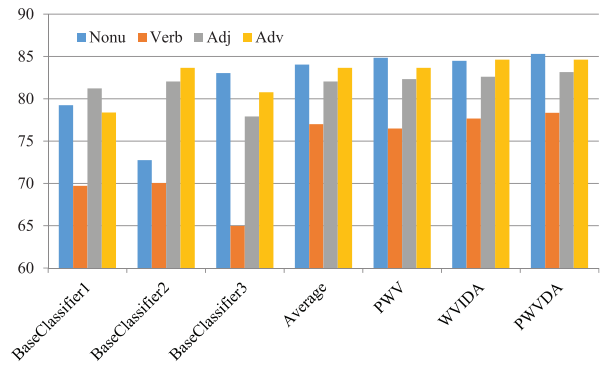

Fig. 3 Effectiveness comparison of different methods on POS (recall, %).

As is shown in Fig. 2, among the four probability-based methods, Average achieves the best performance. Among the four voting-based methods, PWV is the best. The combination effectiveness on coarse-grained sense level is superior to that on fine-grained sense level. Therefore, in the following experiences, we select Average and PWV as the representative methods of probability-based and voting-based methods respectively, compare them with WVIDA and PWVDA on coarse-grained sense level.

5.3.3 Effectiveness Comparison on Different Parts of Speech (POS) and Documents

The recall of three base classifiers, Average, PWV, WVIDA and PWVDA are compared on POS and documents respectively, as are shown in Fig. 3 and Fig. 4.

As is shown in Fig. 3, the following conclusions can be drawn.

- Among the three base classifiers, BaseClassifier1 achieves a moderate performance on each POS, BaseClassifier2 is the best on verb, adjective and adverb, BaseClassifier3 is the best on noun. WSD based on

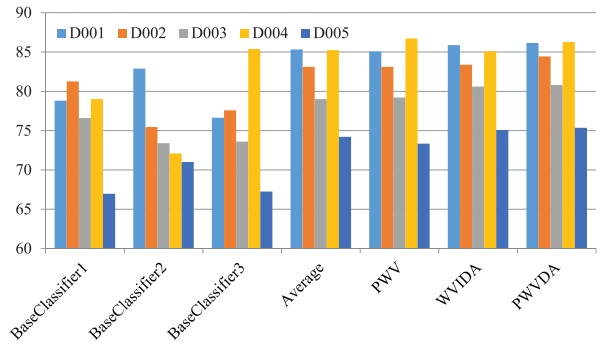

Fig. 4 Effectiveness comparison of different methods on documents (recall, %).

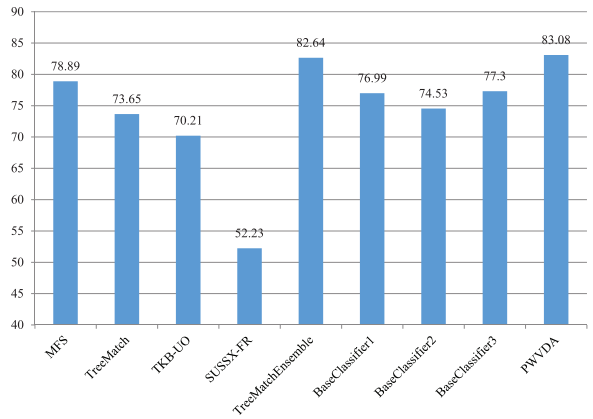

Fig. 5 Effectiveness Comparison of related works (recall, %).

dependency fitness (BaseClassifier2) is good at adjective and adverb, whose effectiveness is obviously better than the other classifiers. Compared with noun and verb, adjective and adverb are more suited with dependency fitness [28]. Graph-based WSD with domain knowledge (BaseClassifier3) utilizes BabelNet [30] as knowledge base. BableNet contains abundant semantic relations, especially among nouns. This may be the reason that BaseClassifier3 achieves the best performance on noun. Because verb is difficult to disambiguate, the three base classifiers' performance on verb is obviously worse than on the others.

- When the four combination methods and the three base classifiers are compared, it is clear that the WSD effectiveness on each POS has been improved by combination methods, especially on verb.
- For combination methods, PWVDA proposed in the paper has achieved the best performance on each POS. The performances of four combination methods on each POS are consistent, which are best on noun, follows on adverb and adjective, and are worst on verb.

As is shown in Fig. 4, the following conclusions can be drawn.

- For the three base classifiers, BaseClassifier1 is the best on D002 and D003, BaseClassifier2 is the best on D001 and D005, BaseClassifier3 is the best on D004.
- When the four combination methods and the three base classifiers are compared, it is clear that WSD effectiveness on each document has been improved greatly.
- For combination methods, Except for D004, PWVDA proposed in the paper has achieved the best performance on the other four documents. The performances of four combination methods on each document are consistent, which are best on D004 and D001, follows on D002 and D003, and are worst on D005.

5.3.4 Comparison with Related Works

In order to further evaluate the performance of PWVDA proposed in the paper, related works with unsupervised and knowledge-based methods on the same dataset (SemEval Task#07) are compared, as is shown in Fig. 5. The related works include MFS, TreeMatch [23], TKB-UO [25], SUSSX-FR [26], TreeMatchEnsemble [24], which are introduced briefly as follows.

- MFS. According to the frequency information in WordNet, the method selects the most common sense as the right sense, which is often as a baseline. Because there is a strong sense skew in a language, MFS is difficult to surpass.
- TreeMatch. The method is based on the fitness of dependency trees. The dependency trees of sense gloss and ambiguous sentence are compared and the sense with the most similarity is selected as right sense.
- TKB-UO. The method is an unsupervised WSD method based on clustering.
- SUSSX-FR. The method is an unsupervised WSD method based on automatically acquired predominant senses.
- TreeMatchEnsemble. The method is a multi-classifier combination method, which combines TreeMatch, Lesk [31] and MFS. If the decision of TreeMatch and Lesk is consistent, their decision is selected as right sense; otherwise, MFS is selected.

As is shown in Fig. 5, the following conclusions can be drawn.

- All individual classifiers, that is TreeMatch, TKB-UO, SUSSX-FX, BaseClassifier1, BaseClassifier2 and BaseClassifier3, fail to surpass MFS. For unsupervised and knowledge-based methods, MFS is still a strong baseline which is difficult to surpass.
- The three base classifiers proposed by us are better than the other individual classifiers, that is, TreeMatch, TKB-UO and SUSSX-FR. In related works, TreeMatch is the best method among unsupervised and knowledge-based methods. This demonstrates the superiority of the three base classifiers proposed by us [27]–[29].
- For the combination methods, PWVDA proposed in the paper is superior to TreeMatchEnsemble (TME). When

TME combines different classifiers, if its two base classifiers fail to get consistent result, it would take MFS as right sense. It is clear that TME depends on MFS excessively, that is the frequency information in WordNet. In contrast, PWVDA fully depends on the base classifiers. It considers the combination strategy from the difference between classifiers' overall performance and the difference between ambiguous instances. This comprehensive consideration makes PWVDA more effective than the other combination methods.

6. Conclusion and Future Work

In the paper, we systematically investigate classifier combination based WSD methods, including probability-based and voting-based approaches. Aiming at the deficiency of existing methods, a novel classifier combination based WSD, that is probability weighted voting method with dynamic self-adaptation, is proposed. Our method considers both of the difference between overall performances of classifiers and the difference between ambiguous instances. On coarse-grained English all-words task in SemEval, our method has shown the superiority over state-of-the-art methods. A series of classifier combination methods are compared on the same dataset, which can provide an effective reference for related research works.

In the future, we will conduct further research to improve our work. Firstly, we will try to optimize the parameters by analyzing the sense distribution, as there is a strong sense-skew in a language. Secondly, we will try to apply our classifier combination method on other applications to verify its effectiveness.

Acknowledgments

The work described in this paper was mainly supported by National Natural Science Foundation of China under Grant 61502259, National Programs for Fundamental Research and Development of China (973 Program) under Grant 2013CB329303 and National Natural Science Foundation of China under Grant 61202244.

References

[1] R. Navigli, "Word sense disambiguation: A survey," ACM Comput. Surv., vol.41, no.2, pp.10:1–10:69, 2009.
[2] A. Raganato, J. Camacho-Collados, and R. Navigli, "Word sense disambiguation: A unified evaluation framework and empirical comparison," Proceeding of 15th Conference of the European Association for Computational Linguistics (EACL 2017), Association for Computational Linguistics, pp.99–110, 2017.
[3] A. Moro, A. Raganato, and R. Navigli, "Entity linking meets word sense disambiguation: a unified approach," Transactions of the Association for Computational Linguistics, vol.2, pp.231–244, 2014.
[4] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On combining classifiers," IEEE Trans. Pattern Anal. Mach. Intell., vol.20, no.3, pp.226–239, 1998.
[5] B. Krawczyk and M. Woźniak, "Untrained weighted classifier combination with embedded ensemble pruning," Neurocomputing, vol.196, pp.14–22, 2016.
[6] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "Drcw-ovo: Distance-based relative competence weighting combination for one-vs-one strategy in multi-class problems," Pattern Recognition, vol.48, no.1, pp.28–42, 2015.
[7] F. Enríquez, F.L. Cruz, F.J. Ortega, C.G. Vallejo, and J.A. Troyano, "A comparative study of classifier combination methods applied to nlp tasks," Information Fusion, vol.14, no.3, pp.255–267, 2013.
[8] R. Florian, S. Cucerzan, C. Schafer, and D. Yarowsky, "Combining classifiers for word sense disambiguation," Natural Language Engineering, vol.8, no.4, pp.327–341, 2002.
[9] M. Carpuat, W. Su, and D. Wu, "Augmenting ensemble classification for word sense disambiguation with a kernel pca model," Proceedings of Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3), pp.88–92, 2004.
[10] D. Klein, K. Toutanova, H.T. Ilhan, S.D. Kamvar, and C.D. Manning, "Combining heterogeneous classifiers for word-sense disambiguation," Proceedings of SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, pp.74–80, Association for Computational Linguistics, 2002.
[11] A.-C. Le, A. Shimazu, V.-N. Huynh, and L.-M. Nguyen, "Semi-supervised learning integrated with classifier combination for word sense disambiguation," Computer Speech and Language, vol.22, no.4, pp.330–345, 2008.
[12] V.-N. Huynh, T.T. Nguyen, and C.A. Le, "Adaptively entropy-based weighting classifiers in combination using dempster–shafer theory for word sense disambiguation," Computer Speech and Language, vol.24, no.3, pp.461–473, 2010.
[13] Y. Zhang and J. Guo, "Word sense disambiguation based on ensembled classifier with dynamic weight adaptation," Journal of Chinese Information Processing, vol.26, pp.3–8, 2012.
[14] Y. Wu, M. Wang, P. Jin, and S. Yu, "Ensembles of classifiers for chinese word sense disambiguation," Journal of Computer Research and Development, vol.45, pp.1354–1361, 2008.
[15] C. Quan, T. He, D. Ji, and S. Yu, "Word sense disambiguation based on multi-ciassifier decision," Journal of Computer Research and Development, vol.43, pp.933–939, 2006.
[16] R. Mihalcea and T. Chklovski, "The senseval-3 english lexical sample task," Proceedings of Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3), pp.88–92, 2004.
[17] A. Kilgarriff and J. Rosenzweig, "Framework and results for english senseval," Computers and the Humanities, vol.34, no.1-2, pp.15–48, 2000.
[18] R. Navigli, K.C. Litkowski, and O. Hargraves, "Semeval-2007 task 07: Coarse-grained english all-words task," Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pp.30–35, Association for Computational Linguistics, 2007.
[19] X. Wang and Y. Matsumoto, "Trajectory based word sense disambiguation," Proceedings of the 20th International Conference on Computational Linguistics (Coling 2004), pp.903–909, Association for Computational Linguistics, 2004.
[20] T. Pedersen, "A simple approach to building ensembles of naive bayesian classiers for word sense disambiguation," Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics, pp.63–69, 2000.
[21] C.A. Le, V.-N. Huynh, A. Shimazu, and Y. Nakamori, "Combining classifiers for word sense disambiguation based on dempster-shafer theory and owa operators," Data & Knowledge Engineering, vol.63, no.2, pp.381–396, 2007.
[22] K. Komiya and M. Okumura, "Automatic domain adaptation for word sense disambiguation based on comparison of multiple classifiers," Proceedings of the 26th Pacific Asia Conference on Language, Information and Computation (PACLIC 26), pp.80–88, 2012.
[23] P. Chen, W. Ding, C. Bowes, and D. Brown, "A fully unsupervised word sense disambiguation method using dependency knowledge,"

Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pp.28–36, Association for Computational Linguistics, 2009.
[24] P. Chen, C. Bowes, W. Ding, and M. Choly, "Word sense disambiguation with automatically acquired knowledge," IEEE Intelligent Systems, vol.27, no.4, pp.46–55, 2012.
[25] H. Anaya-Sánchez, A. Pons-Porrata, and R. Berlanga-Llavori, "Tkbuo: Using sense clustering for wsd," Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pp.322–325, Association for Computational Linguistics, 2007.
[26] R. Koeling and D. McCarthy, "Sussx: Wsd using automatically acquired predominant senses," Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pp.314–317, Association for Computational Linguistics, 2007.
[27] W. Lu, H. Huang, and C. Zhu, "Feature words selection for knowledge-based word sense disambiguation with syntactic parsing," Przeglad Elektrotechniczny, vol.88, pp.82–87, 2012.
[28] W.-P. Lu and H.-Y. Huang, "Word Sense Disambiguation Based on Dependency Fitness with Automatic Knowledge Acquisition," Journal of Software, vol.24, no.10, pp.2300–2311, 2013.
[29] W. Lu, H. Huang, and H. Wu, "Word sense disambiguation with graph model based on domain knowledge," Acta Automatica Sinica, vol.40, pp.2836–2850, 2014.
[30] R. Navigli and S.P. Ponzetto, "BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network," Artificial Intelligence, vol.193, pp.217–250, 2012.
[31] M. Lesk, "Automated sense disambiguation using machine readable dictionaries: How to tell a pine cone from all ice cream cone," Proceedings of the 5th annual international conference on Systems documentation (SIGDOC-86), pp.24–26, Association for Computing Machinery, 1986.

Wenpeng Lu is a full-time associate professor in the School of Information, QiLu University of Technology, China. He received his Ph.D. degree in computer science from Beijing Institute of Technology in 2014. His research areas include natural language processing, artificial intelligence and bioinformatics, especially, word sense disambiguation and sense similarity computation.

Hao Wu is currently a Ph.D. student in the Department of Computer Science and Technology, Beijing Institute of Technology, China. His area of research includes natural language processing and machine translation, especially, sentence similarity computation.

Ping Jian is a full-time lecturer in the Department of Computer Science and Technology, Beijing Institute of Technology, China. She received her Ph.D. degree in pattern recognition and intelligent systems from Graduate University of Chinese Academy of Science in 2010. Her area of research includes natural language syntactic parsing and discourse analysis, which connect to the fields of natural language understanding, machine learning and pattern recognition.

Yonggang Huang is a full-time lecturer in the Department of Computer Science and Technology, Beijing Institute of Technology, China. He received his Ph.D. degree in computer science from Beihang University in 2012. His area of research includes machine learning and its applications.

Heyan Huang is a full-time professor and the dean of Department of Computer Science and Technology, Beijing Institute of Technology. She received her Ph.D. degree in computer science from Institute of Computing Technology, Chinese Academy of Sciences, China, in 1989. Her current research interests include natural language processing, machine translation, social computing.