Word Sense Disambiguation by Combining Classifiers with an Adaptive Selection of Context Representation

Cuong Anh Le,[†] Akira Shimazu[†] and Van-Nam Huynh[††]

Word Sense Disambiguation (WSD) is the task of choosing the right sense of a polysemous word given a context. It is obviously essential for many natural language processing applications such as human-computer communication, machine translation, and information retrieval. In recent years, much attention have been paid to improve the performance of WSD systems by using combination of classifiers. In (Kittler, Hatef, Duin, and Matas 1998), six combination rules including product, sum, max, min, median, and majority voting were derived with a number of strong assumptions, that are unrealistic in many situations and especially in text-related applications. This paper considers a framework of combination strategies based on different representations of context in WSD resulting in these combination rules as well, but without the unrealistic assumptions mentioned above. The experiment was done on four words *interest, line, hard, serve*; on the DSO dataset it showed high accuracies with median and min combination rules.

KeyWords: *Computational Linguistics, Classifier Combination, Word Sense Disambiguation.*

1 Introduction

The automatic disambiguation of word senses has been an interest and concern for many decades. Roughly speaking, word sense disambiguation involves the association of a given word in a text or discourse with a particular sense among numerous potential senses of that word. As mentioned in (Ide and Véronis 1998), this is an "intermediate task" necessary to accomplish most natural language processing tasks. It is obviously essential for language understanding applications, such as message understanding and human-machine communication; it is also at least helpful for other applications whose aim is not language understanding, such as machine translation and information retrieval, among others. Since its inception, many methods involving WSD have been developed in the literature (see, e.g., (Ide and Véronis 1998) for a survey). Practically speaking, an ambiguous word usually has ambiguity regarding its part-of-speech and its meaning. WSD usually considers disambiguating the meaning of a word in a specific part-of-speech. A word in a specific part-of-speech which has several meaning (senses) is called polysemous.

[†] School of Information Science, Japan Advanced Institute of Science and Technology
[††] School of Knowledge Science, Japan Advanced Institute of Science and Technology

Two essential problems are concerned in the task of disambiguating word senses: designing which features are used as evidence for identifying the sense and what learning method is used. Regarding the second problem, during the last decade, many supervised machine learning algorithms have been used for the WSD task, including Naïve Bayesian (NB), decision trees, an exemplar-based, support vector machine, maximum entropy, etc. As observed in studies of machine learning systems, although one of the available learning systems could be chosen to achieve the best performance for a given pattern recognition problem, the set of patterns misclassified by the different classification systems would not necessarily overlap. This means that different classifiers may potentially offer complementary information about patterns to be classified. This observation highly motivated the recent interest in combining classifiers. Especially, classifier combination for lexical disambiguation in WSD has, not surprisingly, received much attention recently from the community, e.g., (Brill and Wu 1998; Kilgarriff and Rosenzweig 2000; Hoste, Hendrickx, Daelemans, and van den Bosch 2002; Pedersen 2000; Klein, Toutanova, Tolga Ilhan, Kamvar, and Manning 2002; Florian, Cucerzan, Schafer, and Yarowsky 2002a; Florian and Yarowsky 2002b; Wang and Matsumoto 2004).

As is well-known, there are basically two classifier combination scenarios. In the first scenario, all classifiers use the same representation of the input pattern. In the second scenario, each classifier uses its own representation of the input pattern. An important application of combining classifiers in this scenario is the possibility of integrating physically different types of features. For determining the sense of a polysemous word, a specific context in which the word appears is given. A set of features extracted from the context will be used as clues for determining an appropriate sense of the target word. Kinds of features usually used include bags of content words, collocations, or some relationship between the target word with surrounding words such as syntactic relation and distance relation. However, utilizing all of these features in a unique set is not always a good idea because each of them, even those of the same kind (for example bag of content words) but with different window sizes, has a different impact on the meaning of the polysemous word, depending on a particular context or on the target word itself. This intuitive observation prompted us to use multi-representation of context as a means of combining individual decisions to reach a consensus.

Concerning the second scenario in combination strategies, (Kittler et al. 1998) presented a theoretical framework for combining classifiers, resulting in commonly-used combination rules including product, sum, max, min, median, and majority voting. First, the product rule was generated as the result of the Bayesian approach, in which distinct pattern representations are used jointly to make a classification decision. Obviously, this rule adopts the assumption of conditional independence between individual classifiers. Furthermore, other rules are generated with two strong assumptions, namely the assumption that posteriori probabilities computed by the respective classifiers will not deviate dramatically from the prior probabilities, and that of equality of priors. However, these assumptions are unrealistic in many situations and especially in text-related applications such as WSD. Therefore, in this paper, we present a new interpretation for obtaining the median, max, min, and majority voting rules without using such assumptions. In addition, some observations from our experiences and other studies such as in (Klein et al. 2002) show that designing features as clues for identifying word sense may be more important than learning methods. In the classifier combination problem in the second scenario, an important thing is to select the various context representations so that the individual classifiers based on them satisfy two criteria: they are "good" classifiers (meaning that the individual classifiers can archive with as high accuracy as possible); and they have rich information so that they can provide complementary information in combination strategies. In this paper we present an ensemble of context representations that tries to fulfill these criteria. Particularly, we experimentally design multiple representations of context, each of which corresponds to an individual classifier, covering enough information to identify the sense of a polysemous word and obtain high accuracies with experiments on the four words *interest*, *line*, *hard*, *serve*, and on the DSO dataset.

The rest of this paper is organized as follows: in the next section related works will be briefly reviewed. Section 3 first introduces WSD with multi-representation of context and explores the product rule of combination. Then, other combination strategies derived from the median combination rule including product, median, max, min, and majority voting are discussed. Section 4 first describes various types of features and different representations of context in previous work, and then presents our selection of context representations for individual classifiers. In Section 5, we present our experimental results and some comparison with previously known results on the same test datasets. Finally, some conclusions are presented in Section 6.

2 Related Work

In this section, we will review previous work related to WSD in the context of using combination methods. As mentioned above, in (Kittler et al. 1998) the combination methods are divided into two main scenarios based on the differences of learning methods and feature sets. In another view, one way to create multiple classifiers is to use subsamples of the training examples. In **bagging**, the training set for each individual classifier is created by randomly drawing training examples with replacement from the initial training set. In **boosting**, the errors made by a classifier learned from a training set are used to construct a new training set in which the misclassified examples get higher weight. By sequentially performing this operation, an ensemble is constructed. One other way to create multiple classifiers is based on multiple feature sets on the same training dataset. Methods of combining the outputs of component classifiers in an ensemble include **simple voting**, wherein each component classifier gets an equal vote, and **weighted voting**, in which each component classifier's vote is weighted by its accuracy. The most interesting approach to combination is **stacking**, in which a classifier is trained to predict the correct output class when given as input the outputs of the ensemble classifiers, and possibly additional information.

In the WSD literature, the first empirical study of combining classifiers was presented in (Kilgarriff and Rosenzweig 2000), in which the authors combined the output of the participating SENSEVAL1 systems via simple voting. (Pedersen 2000) built an ensemble of Naive Bayesian classifiers, each of which is based on lexical features that represent co-occurring words in varying sized windows of context. (Klein et al. 2002) use a stacking type of combination techniques. First, individual classifiers were constructed based on different training datasets and learning methods, and then they were ranked according to results obtained from testing on held-out data. In the next step, majority voting, weighted voting, and maximum entropy were used as combination strategies. (Hoste et al. 2002) used word experts consisting of four memory-based learners trained on different context. Output of the word experts is based on majority voting or weighted voting. In (Florian et al. 2002a) the authors used six different classifiers as components of their combination. They compared several different combination strategies which include combining the posterior distribution, combination based on order statistics, and several different voting strategies. (Frank, Hall, and Pfahringer 2003) presented a locally weighted Naive Bayesian model. For a given test instance, they first chose k-nearest neighbors from training samples, then constructed a Naive Bayesian classifier by using these k-nearest neighbors instead of all training samples. (Wang and Matsumoto 2004) presented a kind of stacking; individual classifiers were built using NB with varying sized windows of context that are similar to Pedersen's approach (Pedersen 2000), and then used K-nearest neighbors as the meta learning method.

Several combination approaches in WSD, most of which used majority voting or weighted voting on the output of individual classifiers, were based on different sets of features or different learning methods. Some of them proposed different approaches such as using maximum entropy (Klein et al. 2002), or a stacking method (Wang and Matsumoto 2004) as combination strategies. Linear combination strategies as shown in the next section have not yet been considered in WSD studies, with the exception of the recently report (Le, Huynh, and Shimazu 2005), but in which the authors simply applied the framework suggested in (Kittler et al. 1998). Furthermore, from our observation, and also as shown by others such as (Klein et al. 2002), differences in representations of context have a stronger influence than differences in learning algorithms. Therefore a selection of context representation for individual classifiers adaptive with combination strategies may play an important role in obtaining high accuracy, which was lacking in previous works.

3 WSD with Multi-representation and Combination Strategies

In this section, we first observe that the given context of a polysemous word can be represented in different ways, so that each of them can be used to build an individual classifier. It is well-known that in the WSD problem, context plays an essentially important role to identify the meaning of a polysemous word. Given an ambiguous word w, which may have m possible senses (classes): $\{\omega_1, \omega_2, \ldots, \omega_c\}$, in a context C, the task is to determine the most appropriate sense of w.

Generally, context C can be used in two ways (Ide and Véronis 1998): in the *bag-of-words approach*, the context is considered as a bag of words in some window surrounding the target word w; in the *relational information based approach*, the context is considered in terms of some relation to the target such as distance from the target, syntactic relations, selectional preferences, phrasal collocation, semantic categories, etc. Thus, for a target word w, we may have different representations of context C corresponding to different views of the context. Assume we have R representations of C, say $\mathbf{f}_1, \ldots, \mathbf{f}_R$, serving to identify the right sense of the target w. Clearly, each $\mathbf{f}_i$ can be also considered as a semantical representation of w. Each representation $\mathbf{f}_i$ of the context has its own type depending on which way the context is used (for details, see Section 3). In the sequence, we can use a set of features and a representation interchangeably without danger of confusion. It is quite natural to assume that there are R classifiers, each representing the context by a distinct set of features. The set of features $\mathbf{f}_i$, which is considered as a representation of context C of the target w, is used by the i-th classifier.

In the remainder of this section, we first present the *product* rule based on the Bayesian approach, which is the same as in (Kittler et al. 1998). Next, a basic framework for combining classifiers and the *median* rule are presented, and then other combination rules including *max*, *min*, and *majority voting* are derived from the *median* rule using lower and upper approximations. It is worth emphasizing again that our approach is different from (Kittler et al. 1998) in which the authors derived the min rule from product rule, and the sum rule was yielded by the product rule with the assumption that posteriori probabilities computed by the respective classifiers will not deviate dramatically from prior probabilities. Other rules including median, max, and min rules were derived from sum rule with the assumption of equality of priors and some approximations.

3.1 Product Rule

Under a mutually exclusive assumption, given representations $\mathbf{f}_i$ $(i = 1, \ldots, R)$, the Bayesian theory suggests that the word w should be assigned to class ω_k provided the a posteriori probability of that class is maximum, namely

$$k = \arg\max_j P(\omega_j | \mathbf{f}_1, \ldots, \mathbf{f}_R) \tag{1}$$

That is, in order to utilize all the available information to reach a decision, it is essential to consider all the representations of the target simultaneously.

The decision rule (1) can be rewritten using Bayes theorem as follows:

$$k = \arg\max_j \frac{P(\mathbf{f}_1, \ldots, \mathbf{f}_R | \omega_j) P(\omega_j)}{P(\mathbf{f}_1, \ldots, \mathbf{f}_R)}$$

Because the value of $P(\mathbf{f}_1, \ldots, \mathbf{f}_R)$ is unchanged with variance of ω_j, we have

$$k = \arg\max_j P(\mathbf{f}_1, \ldots, \mathbf{f}_R | \omega_j) P(\omega_j) \tag{2}$$

As we see, $P(\mathbf{f}_1, \ldots, \mathbf{f}_R | \omega_j)$ represents the joint probability distribution of the representations extracted by the classifiers. Assume that the representations used are conditionally independent, so that the decision rule (2) can be rewritten as follows:

$$k = \arg\max_j P(\omega_j) \prod_{i=1}^{R} P(\mathbf{f}_i | \omega_j) \tag{3}$$

According to Bayes rule, we have:

$$P(\mathbf{f}_i | \omega_j) = \frac{P(\omega_j | \mathbf{f}_i) P(\mathbf{f}_i)}{P(\omega_j)} \tag{4}$$

Substituting (4) into (3), we obtain:

$$k = \arg\max_j P(\omega_j) \prod_{i=1}^{R} \frac{P(\omega_j | \mathbf{f}_i) P(\mathbf{f}_i)}{P(\omega_j)} = \arg\max_j [P(\omega_j)]^{-(R-1)} \prod_{i=1}^{R} P(\omega_j | \mathbf{f}_i) \tag{5}$$

The decision rule (5) quantifies the likelihood of a hypothesis by combining the a posteriori probabilities generated by the individual classifiers by means of a product rule.

3.2 Derived Combination Strategies

Let $D = \{D_1, \dots, D_R\}$ be a set of classifiers, and let $\Omega = \{\omega_1, \dots, \omega_c\}$ be a set of class labels. Each classifier gets as its input a representation of polysemous word w and assigns it to a class label from Ω. Alternatively, we may define the classifier output to be a c-dimensional vector

$$D_i(w) = [d_{i,1}(w), \dots, d_{i,c}(w)] \tag{6}$$

where $d_{i,j}(w)$ is the degree of "support" given by classifier D_i to the hypothesis that w comes from class ω_j. Most often $d_{i,j}(w)$ is an estimation of the posterior probability $P(\omega_i|w)$. In fact, the detailed interpretation of $d_{i,j}(w)$ beyond a "degree support" is not important for the operation for any of the combination methods studies here.

With the notation $\mathbf{f}$ as a set of the different context representations $\{\mathbf{f}_1, \dots, \mathbf{f}_R\}$, it is convenient to organize the output of all R classifiers based on $\mathbf{f}$ in a decision matrix as follows.

$$DP(\mathbf{f}) = \begin{bmatrix} d_{1,1}(w) & \dots & d_{1,j}(w) & \dots & d_{1,c}(w) \\ \dots & & & & \\ d_{i,1}(w) & \dots & d_{i,j}(w) & \dots & d_{i,c}(w) \\ \dots & & & & \\ d_{R,1}(w) & \dots & d_{R,j}(w) & \dots & d_{R,c}(w) \end{bmatrix} \tag{7}$$

Thus, the output of classifier D_i is the i-th row of the decision matrix, and the support for class ω_j is the jth column. *Combining classifiers* means to find a class label for $\mathbf{f}$ based on the R classifiers outputs. We look for a vector with c final degrees of support for the classes, denoted

$$D(\mathbf{f}) = [\mu_1(w), \dots, \mu_c(w)] \tag{8}$$

If a single class label of w is needed, we use the maximum membership rule: Assign w to class ω_s iff

$$\mu_s(w) \geq \mu_t(w), \forall t = 1, \dots, c. \tag{9}$$

Note that, returning to the Product rule (5), we have

$$d_{i,j}(w) = P(\omega_j|\mathbf{f}_i), \text{ for } i = 1, \dots, R; j = 1, \dots, c;$$

$$\mu_j = [P(\omega_j)]^{-(R-1)} \prod_{i=1}^{R} P(\omega_j|\mathbf{f}_i), \text{ for } j = 1, \ldots, c$$

where $\mathbf{f}_i$ is a representation of w

Median Rule.

Let us return to the decision matrix (7), each classifier D_i supports a degree $d_{i,j}(w)$ for the class ω_j. According to (Perrone and Cooper 1993), if the errors made by R classifiers D_i, $i = 1, \ldots, R$, are uncorrelated and unbiased, then these R classifiers can be combined into a classifier that supports the class ω_j with the degree

$$\mu_j = \left[\frac{1}{R}\sum_{i=1}^{R} d_{i,j}(w)\right] \tag{10}$$

Let $d_{i,j}(w)$ be an estimation of the posterior probability $P(\omega_i|w)$. Noting that $\mathbf{f}_i$ is the representation of w with respect to the classifier D_i, equation (10) then becomes

$$\mu_j = \left[\frac{1}{R}\sum_{i=1}^{R} P(\omega_j|\mathbf{f}_i)\right] \tag{11}$$

Therefore, the class (sense) ω_k is chosen as the best class for the target word under the median rule as follows

$$k = \arg\max_j \left[\frac{1}{R}\sum_{i=1}^{R} P(\omega_j|\mathbf{f}_i)\right] \tag{12}$$

Max and Min Rule.

From the median rule, it is interesting that the max and min rules can be derived using the following inequality

$$\min_{i=1}^{R} P(\omega_j|\mathbf{f}_i) \leq \frac{1}{R}\sum_{i=1}^{R} P(\omega_j|\mathbf{f}_i) \leq \max_{i=1}^{R} P(\omega_j|\mathbf{f}_i) \tag{13}$$

This relationship suggest that the median combination rule can be approximated by the above upper and lower bounds, as appropriate. Starting from (11) and maximizing the sum by the upper bound, we obtain the Max rule

$$k = \arg\max_j \left[\max_{i=1}^{R} P(\omega_j|\mathbf{f}_i)\right] \tag{14}$$

Also, minimizing the sum (11) by the lower bound, we obtain the Min rule

$$k = \arg\max_j \left[\min_{i=1}^{R} P(\omega_j|\mathbf{f}_i)\right] \tag{15}$$

Furthermore, the max combination rule can be interpreted in a intuitive way: for the class ω_j and an input word w, each classifier D_i supports the class ω_j with a degree $d_{i,j}(w)$, so with the hypothesis that *there exist* one classifier which supports ω_j, the maximum of these supported degrees $(d_{i,j}(w), i = 1, \ldots, c)$ is considered as the global support of class ω_j. Using the maximum membership rule we generate the max combination rule.

For the min combination, with the hypothesis that *all* classifiers support class ω_j, so the minimum of degrees $(d_{i,j}(w), j = 1, \ldots, c)$ will be chosen as the global support for class ω_j. The class corresponding to the maximum of the global support degrees is chosen.

Majority Voting.

Majority voting follows a simple rule as: it will vote for the class which is chosen by maximal number of individual classifiers. Suppose that the classifier D_i chooses class ω_k as the final decision, then we can consider the outputs of D_i as follows:

$$d_{i,j}(w) = \begin{cases} 1, & \text{if } j = k \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

Still considering the output of each classifier D_i by the posterior, it is natural to use the maximum membership rule, so formula (16) can be rewritten as

$$d_{i,j}(w) = \begin{cases} 1, & \text{if } P(\omega_j|\mathbf{f}_i) = \max_k P(\omega_k|\mathbf{f}_i) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

Substituting (17) into (10), we have majority voting: the right class (sense) ω_k is determined as follows:

$$k = \arg\max_j \sum_i d_{i,j}(w) \tag{18}$$

4 Representation of Context

As mentioned previously, context plays an essential role in WSD. Selection of an effective representation of context may be more important than the learning algorithm. Several kinds of information are usually used to predict the senses of a word. Among them, information about topic context, which is represented by a bag of words, is always used in WSD studies. (Ng and Lee 1996) proposed the use of more linguistic knowledge resources including topic context, collocation of words, and the verb-object syntactic relationship, which then became the popular template of knowledge used in many studies. (Leacock, Chodorow, and Miller 1998) used another type of information, which includes words or part-of-speech tags assigned with their positions in relation to the target word. In (Le and Shimazu 2004; Le et al. 2005), Le et al. used five kinds of information including bag of content words, collocation of words, collocation of part-of-speech tags, words assigned with their position, and part-of-speech assigned with their position. All these kinds of features can be grouped into a unique set of features and used in a learning algorithm. However, each kind of knowledge has a different effect on the decision to determine the right sense of a polysemous word. In some cases, only information about collocation can discriminate word senses. In other cases, information about topic context is enough for that task, and even that topic context with different window sizes will have causes different effects on the decision. Therefore, it is of interest that distinct representations of context can be used jointly to identify the meaning of the target word based on combination strategies.

4.1 Representations in previous works

In the literature related to combining classifiers for WSD based on different sets of features, only topic context with different sizes of context windows is used to create different representations of a polysemous word, such as in (Pedersen 2000; Wang and Matsumoto 2004). In this work, we do not consider other information than the orthographic form of words and part-of-speech tags. For easy understanding of what constitutes features, we define that: w_i is the word at position i in the context of the ambiguous word w and p_i is the part-of-speech tag of w_i, with the convention that the target word w appears precisely at position 0 and i will be negative (positive) if w_i appears on the left (right) of w.

(Pedersen 2000) considered several context windows on both the left and the right and grouped them into three kinds: small with window sizes 0, 1, 2; medium with window sizes 3, 4, 5; and large with window sizes 10, 25, 50. There were 81 different representations generated from combining between left and right window sizes. He then chose the best of each kind for the majority voting procedure. (Wang and Matsumoto 2004) also used only the content words in various window sizes with different left and right window sizes was (1,2,3,4,5,6,10,15,20). In (Le et al. 2005), the authors tested the combination strategies on two types of context representations. In the first type, they borrowed this feature space division from Pedersen and used the maximum window size in each kind, consequently nine different representations were generated based on nine different combinations of left and right windows: (2, 2), (2, 5), (2, 50), (5, 2), (5, 5), (5, 50), (50, 2), (50, 5), and (50, 50). In the second type they used five context representations corresponding to five kinds of features including bag of content words, collocation of words (and part-of-speech tags), and words (and part-of-speech tags) assigned with their positions.

4.2 An adaptive selection of context-representation ensemble

The combination strategies to be considered include product, median, max, min rule, and majority voting. As mentioned in Section 3, the product rule is derived based on Bayes theory with the independent assumption of context representations, so that the context representations are to be built to satisfy as much as possible two criteria: they are mutually exclusive or in other words, they are independent; and the combination of the context representations contains as much rich information as possible. For the remaining combination rules, the context representations need to contain some important characteristics. They should be based on different kinds of information, so that the corresponding classifiers make different errors and therefore can supply complementary information. However, they do not need to satisfy the assumption of independence. We design them so that the corresponding individual classifiers are as good as possible, and thus they can make the combination more efficient. It is worth emphasizing that two of the most important kinds of information for determining the sense of a polysemous word include the topic of the given context and relational information representing the structural relations between the target word and the surrounding words in a local context.

We now discuss which features will represent each of those kinds of information. First of all, the topic of the context can be represented by a set of content words in a context window. In almost WSD studies, the window size is 50. However, as mentioned above, in some cases and depending on the particular word, the window size needs to be smaller or large. Therefore, we discriminate the context window with left and right sizes. We design two sizes of windows including a local size (5) and a large size (25). Consequently, four different sets of topic features are created by combining the left and right sizes, including: (5,5), (5,25), (25,5), and (25,25). Another feature, collocation, is a knowledge resource which has been described as the most informative resource for determining word sense (Ng and Lee 1996). Our observations have also shown that collocation is a very useful features containing information about relationship between the target word and its neighbors. In summary, there are a total of five sets of features will be used in our work, concretely as follows:

- s_1 is a set of collocations of words with collocation lengths consisting of 1, 2, and 3 (not counting the target word):

$$s_1 = \{w_{-3}w_{-2}w_{-1}w_0, w_{-2}w_{-1}w_0w_1, w_{-1}w_0w_1w_2, w_0w_1w_2w_3,$$

$$w_{-2}w_{-1}w_0, w_{-1}w_0w_1, w_0w_1w_2, w_{-1}w_0, w_0w_1\}$$

- s_2 is a set of unordered words in window size (5,5):

$$s_2 = \{w_{-5}, \ldots, w_{-2}, w_{-1}, w_1, w_2, \ldots, w_5\}$$

- s_3 is a set of unordered words in window size (5,25):

$$s_3 = \{w_{-5}, \ldots, w_{-2}, w_{-1}, w_1, w_2, \ldots, w_{25}\}$$

- s_4 is a set of unordered words in window size (25,5):

$$s_4 = \{w_{-25}, \ldots, w_{-2}, w_{-1}, w_1, w_2, \ldots, w_5\}$$

- s_5 is a set of unordered words in window size (25,25):

$$s_5 = \{w_{-25}, \ldots, w_{-2}, w_{-1}, w_1, w_2, \ldots, w_{25}\}$$

In the experiment, we constructed two sets of context representation. The first set supports the independent assumption, so we designed each representation as a feature set listed above: $R_1 = \{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5\}$, where $\mathbf{f}_i = s_i$ for $i = 1, 2, 3, 4, 5$.

In the second one, we tried to design a set of classifiers that can utilize the power of combination strategies median, max, min, and majority voting, which do not depend on the independent assumption. As shown in many WSD studies, collocation is the most important information for most ambiguous words, and our experiment also shows that an individual classifier will become more efficient if it is based on the feature set containing collocations, and consequently it makes better combinations. Such an observation suggest to us that we design the representations with overlapping sets of collocations. Therefore we construct the five context representations as follow: $R_2 = \{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5\}$ where

- $\mathbf{f}_1 = \{s_1\}$
- $\mathbf{f}_2 = \{s_1 \cup s_2\}$
- $\mathbf{f}_3 = \{s_1 \cup s_3\}$
- $\mathbf{f}_4 = \{s_1 \cup s_4\}$
- $\mathbf{f}_5 = \{s_1 \cup s_5\}$

5 Experiments

In our experiments, each individual classifier is a naive Bayesian classifier built on a context representation. We have five individual classifiers corresponding to five context representations in two models, R_1 and R_2. They will jointly make a consensus decision under the combination rules: product, median, max, min, and majority voting. The remainder of this section first presents the computation of posterior probability $P(\omega_j|\mathbf{f}_i)$, and then presents test data, results, and some discussion.

5.1 Computing probabilities

We assume that in all combination strategies, the support degree $d_{i,j}(w)$ is estimated by posterior probability $P(\omega_j|\mathbf{f}_i)$, for $j = 1, \ldots, c$; $i = 1, \ldots, R$. For the context C, suppose that the representation $\mathbf{f}_i$ of C is represented by a set of features $\mathbf{f}_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,n_i})$, and that the features $f_{i,j}$ are conditionally independent. According to Bayes theory, we have:

$$P(\omega_j|\mathbf{f}_i) = \frac{P(\omega_j)P(\mathbf{f}_i|\omega_j)}{P(\mathbf{f}_i)} = \frac{P(\omega_j)\prod_{k=1}^{n_i} P(f_{i,k}|\omega_j)}{P(\mathbf{f}_i)} \tag{19}$$

For simplicity, assume that we are working on the representation $\mathbf{f}_i$, we then have

$$\sum_{j=1}^{c} P(\omega_j|\mathbf{f}_i) = 1$$

Let us denote

$$r_j = \frac{P(\omega_j|\mathbf{f}_i)}{P(\omega_1|\mathbf{f}_i)}, \text{ for } j = 1, \ldots, c$$

With this notation, we immediately obtain

$$P(\omega_1|\mathbf{f}_i) = \frac{1}{\sum_{j=1}^{c} r_j} \tag{20}$$

Clearly, $r_1 = 1$. We will then compute r_j ($j = 2, \ldots, c$) based on the following formulation. From (19), we have

$$r_j = \frac{P(\omega_j|\mathbf{f}_i)}{P(\omega_1|\mathbf{f}_i)} = \frac{P(\omega_j)\prod_{k=1}^{n_i} P(f_{i,k}|\omega_j)}{P(\omega_1)\prod_{k=1}^{n_i} P(f_{i,k}|\omega_1)}$$

Taking the log of the last expression, we obtain

$$\log(r_j) = \sum_{k=1}^{n_i} \log(P(f_{i,k}|\omega_j)) + \log(P(\omega_j)) - \sum_{k=1}^{n_i} \log(P(f_{i,k}|\omega_1)) - \log(P(\omega_1)) \tag{21}$$

which is easy to compute more exactly. Once all r_j are computed via (21), it is easily to derive probabilities $P(\omega_j|\mathbf{f}_i)$, for $j = 1, \ldots, c$, from (20).

The probability of sense ω_j, $P(\omega_j)$, and the conditional probability of a feature $f_{i,k}$ given the sense ω_j, $P(f_{i,k}|\omega_j)$, are computed via maximum-likelihood estimation as:

$$P(\omega_j) = \frac{\text{count}(\omega_\text{j})}{N}$$

and

$$P(f_{i,k}|\omega_j) = \frac{\mathrm{count(f_{i,k}, \omega_j)}}{\mathrm{count(\omega_j)}}$$

where $\mathrm{count(f_{i,k}, \omega_j)}$ is the number of occurrences of $f_{i,k}$ in a context of sense ω_j in the training corpus, $\mathrm{count(\omega_j)}$ is the number of occurrences of ω_j in the training corpus, and N is the total number of occurrences of the polysemous word w or the size of the training dataset. To avoid the effects of zero counts when estimating the conditional probabilities of the model, when meeting a new feature $f_{i,k}$ in a context of the test dataset, for each sense ω_j we set $P(f_{i,k}|\omega_j)$ equal to $\frac{1}{N}$.

5.2 Data and Results

We tested on the datasets for four words, namely *interest*, *line*, *serve*, and *hard*, which are used in numerous comparative studies of word sense disambiguation methodologies such as (Pedersen 2000; Ng and Lee 1996; Bruce,and Wiebe 1994; Leacock et al. 1998). We obtained those datasets from Pedersen's homepage [1]. There are 2369 instances of *interest* with 6 senses, 4143 instances of *line* with 6 senses, 4378 instances of *serve* with 4 senses, and 4342 instances of *hard* with 3 senses. For evaluating on a large dataset, we tested the DSO corpus published in (Ng and Lee 1996), which contains 192,800 semantically annotated occurrences of 121 nouns and 70 verbs corresponding to the most frequently used and ambiguous English words.

In the experiment, a 10-fold cross validation was used. Table 1 and Table 2 show results when testing on the four words with representations R_1 and R_2 respectively. Their columns include results from separately testing five individual classifiers $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_5\}$, and the next column contains maximum results from the five individual classifiers. The next columns are results of using the various combination rules including product, median, max, min, and majority voting, and the final column is the maximum result from the five combination rules. From the experiment on the DSO dataset, like (Escudero, Marquez, and Rigau 2000), we show results of the 15 most frequent words in DSO in Table 3 and Table 4. The columns of these tables are the same as Table 1 and Table 2.

Some conclusions are extracted from these tables as follows.

- The new selection of context presentations, that accepts overlapping of collocations in each presentation, makes the individual classifiers more efficient and help us to obtain better results on combination rules. We can see that the results obtained from representation R_2 are better than those from representation R_1 for most test words when

[1] http://www.d.umn.edu/~tpederse/data.html

	f1	f2	f3	f4	f5	max f	product	median	max	min	m-vote	max combination
interest	89.2	88.7	85.7	86.2	85.2	89.2	91.8	90.9	91.1	91.6	90.2	91.6
line	78.3	83.9	84.1	84.1	86.2	86.2	92.2	91.1	90.4	91.4	90.3	92.2
hard	91.0	90.5	88.3	87.1	85.3	91.0	92.4	91.3	92.4	92.2	91.3	92.4
server	80.5	88.7	86.3	83.3	83.6	88.7	90.5	90.4	90.1	90.5	90.4	90.5

Table 1 Result with representation R1 for the four words.

	f1	f2	f3	f4	f5	max f	product	median	max	min	m-vote	max combination
interest	89.1	93.0	92.5	92.5	92.5	93.0	93.9	94.0	93.9	93.8	93.8	94.0
line	77.9	87.1	88.7	90.2	91.1	91.1	91.9	91.4	91.6	91.5	90.9	91.9
hard	90.8	92.7	92.4	91.8	92.3	92.7	92.3	92.9	92.7	92.7	92.9	92.9
serve	80.6	91.5	90.6	89.4	89.4	91.5	91.3	91.6	91.8	91.6	91.3	91.8

Table 2 Result with representation R2 for the four words.

compared in the same combination rules (the same columns).

- In average, the min and median rules show the best results, and for DSO data, product is shown as the worse combination rule.
- Although the independent assumption is violated, the result of product rule of individual classifiers based on feature sets overlapped with rich features such as collocations can improve the accuracy.

Table 5 compares our proposed method with (Ng and Lee 1996), (Escudero, Marquez, and Rigau 2000a), and (Le and Shimazu 2004). It shows our approach achieves better results with the min and median combination rule. Other information for comparison can be found in Tables 6 and 7. Table 6 presents the comparison of the min rule in our method with various WSD studies that also tested on the four words *interest*, *line*, *hard*, and *serve*. Table 7 compares Escudero et al. (Escudero et al. 2000), (Le and Shimazu 2004) and the result from our min combination rule test on the 15 most frequent words which appear in DSO data. These tables show that by using the min combination rule and the our context representation, we will obtain the better results. It also emphasizes that our new ensemble of context representations is stronger than previous representations.

6 Conclusion

In this paper we first argued that various ways of using context in WSD can be considered as distinct representations of a polysemous word, and then, that these representations can

	f1	f2	f3	f4	f5	max f	product	median	max	min	voting	max combination
age	72.9	69.6	67.2	71.7	72.5	72.9	74.7	77.0	76.4	77.2	75.4	77.2
art	60.4	57.8	62.1	61.1	61.8	62.1	65.3	64.8	68.5	66.3	64.8	68.5
car	94.3	95.0	95.4	95.3	95.9	95.9	86.1	96.0	96.2	96.3	96.2	96.3
child	90.4	83.2	80.0	74.3	74.5	90.4	74.1	85.1	87.5	87.8	84.9	87.8
church	65.5	75.5	74.6	74.9	76.8	76.8	77.6	77.9	75.7	78.4	78.2	77.9
cost	78.1	83.7	82.4	85.7	85.3	85.7	83.3	87.3	86.7	86.9	87.1	87.3
fall	74.7	79.2	78.8	76.6	77.5	79.2	81.2	82.3	81.1	81.8	81.5	82.3
head	76.0	78.3	77.2	78.5	76.2	78.3	72.3	82.2	80.7	80.9	81.4	82.2
interest	68.1	69.2	68.4	67.6	69.2	69.2	73.4	71.9	71.3	72.7	72.1	73.4
know	46.3	46.1	45.3	44.5	44.7	46.3	53.8	52.3	51.3	54.1	49.7	54.1
line	51.5	52.6	51.2	51.4	51.5	52.6	54.6	58.5	59.0	58.4	57.3	59.0
set	50.7	52.6	49.9	51.3	49.9	52.6	46.4	56.7	56.5	56.8	56.5	56.8
speak	67.9	69.7	67.9	66.9	66.7	69.7	69.7	74.0	73.8	76.3	74.4	76.3
take	40.1	45.7	41.9	41.9	39.9	45.7	28.8	48.0	45.5	45.4	46.5	46.5
work	53.0	53.8	53.8	53.6	55.0	55.0	59.8	59.9	59.3	59.7	59.7	59.9
Average	70.0	67.5	66.4	66.3	66.5		66.7	71.6	71.3	71.9	71.0	

Table 3 Result with representation R1 for the 15 words in DSO.

	f1	f2	f3	f4	f5	max f	product	median	max	min	voting	max combination
age	72.3	74.1	73.5	75.5	74.9	75.5	68.4	77.0	75.8	76.2	76.6	77.0
art	59.6	63.1	66.3	69.1	69.6	69.6	66.6	66.6	66.3	66.8	66.1	66.8
car	94.1	94.8	95.6	95.4	96.2	96.2	85.6	96.2	96.5	96.5	96.2	96.5
child	89.5	89.1	87.8	84.8	84.8	89.5	75.9	88.2	87.2	87.3	88.7	88.7
church	66.9	74.7	77.6	77.1	77.9	77.9	79.5	79.8	80.0	80.0	79.8	80.0
cost	77.8	80.8	81.8	83.6	84.1	84.1	79.5	83.3	84.0	84.1	83.2	84.1
fall	75.3	80.7	82.0	80.6	81.8	82.0	80.6	82.3	82.6	82.7	82.1	82.7
head	74.9	78.9	80.0	79.3	79.6	80.0	80.0	82.3	81.3	81.5	81.7	82.3
interest	68.2	72.0	71.9	72.1	72.9	72.9	73.3	73.3	73.0	73.2	73.3	73.3
know	46.2	51.1	52.7	51.2	52.1	52.7	49.0	53.7	53.1	54.2	52.0	54.2
line	50.9	59.2	60.2	60.2	60.6	60.6	57.3	62.8	62.7	62.7	61.9	62.8
set	52.0	55.9	56.1	57.5	57.7	57.7	51.0	59.0	58.2	59.7	59.2	59.7
speak	69.9	71.2	71.8	72.6	72.2	72.6	69.1	73.6	73.2	74.0	73.6	74.0
take	40.5	47.7	47.4	45.9	45.3	45.9	32.5	47.9	47.1	47.3	47.4	47.9
work	51.7	57.2	58.0	58.2	59.9	59.9	60.4	61.7	60.8	61.9	59.8	61.9
Average	66.0	70.0	70.8	70.9	71.3		67.2	72.5	72.1	72.5	72.1	

Table 4 Result with representation R2 for the 15 words in DSO.

be jointly used to identify the meaning of the target word. This consideration allowed the application of a common theoretical framework for combining classifiers, developed in (Kittler et al. 1998), to create numerous strategies of classifier combination for WSD. However, since some of the strong assumptions used in (Kittler et al. 1998) to yield the combination rules

	NL	Es	LS	combination rules				
				product	median	max	min	majority vote
Nouns(121)	-	70.8	72.7	70.3	73.1	72.9	73.1	72.7
Verbs(70)	-	67.5	66.4	59.7	66.8	66.4	66.9	66.5
Average	68.6	69.5	70.4	66.4	70.8	70.5	70.8	70.4

Table 5 Result with representation R_2 comparing with other studies on DSO, where NL, Es, and LS are abbreviation for (Ng and Lee 1996), (Escudero et al. 2000) and (Le and Shimazu 2004), respectively.

	BW	M	NL	LC	P	LS	min rule (R2)
interest	78	–	87	–	89	91.4	93.8
line	–	72	–	84	88	89.4	91.4
hard	–	–	–	83	–	91.0	92.2
serve	–	–	–	83	–	89.6	90.5

Table 6 The comparison between the min rule on representation R2 with previous studies where BW, M, NL, LC, P, and LS are abbreviation for (Bruce and Wiebe 1994), (Mooney 1996), (Ng and Lee 1996), (Leacock et al. 1998), (Pedersen 2000), and the best results from the combination rules in (Le et al. 2005), respectively.

are not suitable for text-related applications, particularly WSD, we have developed a new combination framework for generating these combination rules including median, max, min, and majority rule; the product was yielded in the same way as in (Kittler et al. 1998). This interpretation allowed us to construct the ensemble of context representations which accepted some overlapping features. The context representations were built using collocations and contents words in different windows of the target word. Two ensembles of context representations have been designed: one is without overlapping features (considered to satisfy independent assumption); in the other, all representations contain collocation features. The combination rules with an individual classifier based on two models of context representations were tested on the DSO corpus and on the four words - namely *interest*, *line*, *serve*, and *hard*, - which are used in numerous comparative studies of word sense disambiguation methodologies. The experiment showed that with these context representations, combining classifiers improves the accuracy of WSD specially with min and median combination rules. In addition, designing an ensemble of context representations in model R2 gives high accuracy. Comparing with other studies on the same test data, it has been shown that our approach is promising.

As future work, we plan to apply this framework of classifier combination with classifiers

	Number of examples/senses	Escudero et al.	Le and Shimazu	Our-Min
age(n)	493/4	74.7	73.9	76.2
art(n)	405/5	57.5	68.0	66.8
car(n)	1381/5	96.8	96.0	96.5
child(n)	1068/4	92.8	87.3	87.3
church(n)	373/4	66.2	76.0	80.0
cost(n)	1500/3	87.1	84.3	84.1
fall(v)	1500/19	81.1	83.5	82.7
head(n)	870/14	79.0	80.7	81.5
interest(n)	1500/7	65.4	73.5	73.2
know(v)	1500/8	48.7	51.9	54.2
line(n)	1342/26	54.8	63.6	62.7
set(v)	1311/19	55.8	59.1	59.7
speak(v)	517/5	72.2	68.9	74.4
take(v)	1500/30	46.7	47.7	47.3
work(n)	1469/7	50.7	61.1	61.9
Avg.		68.6	71.7	72.5

Table 7 This table borrowed a part from (Le and Shimazu 2004) shows the comparison between Escudero et al. (Escudero et al. 2000), Le and Shimazu (Le and Shimazu 2004), and our method with min combination rule notated by "Our-Min", on the 15 most frequent words in DSO

which are based on different learning methods such as example-based, maximum entropy, and support vector machine. In addition, strategies of weighted combination of classifiers would be interested to consider in the spirit of this framework.

Reference

Brill, E., and Wu, J. (1998). "Classifier combination for improved lexical disambiguation." *In Proceedings of COLING-ACL'98*, pp. 191–195.

Bruce, R., and Wiebe, J.(1994). "Word-Sense Disambiguation using Decomposable Models." *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 139–145.

Escudero, G., Màrquez, L., and Rigau, G. (2000). "Naive Bayes and exemplar-based approaches to Word Sense Disambiguation revisited." *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*, pp. 421–425.

Escudero, G., Màrquez, L., and Rigau, G. (2000). "Boosting applied to Word Sense Disambiguation." *Proceedings of the 11th European Conference on Machine Learning (ECML)*, pp. 129–141.

Florian, R., Cucerzan, S., Schafer, C., and Yarowsky, D. (2002). "Combining Classifiers for

Word Sense Disambiguation." *Journal of Natural Language Engineering*, *8* (4). pp. 327–341

Florian, R., and Yarowsky, D. (2002). "Modeling consensus: Classifier combination for Word Sense Disambiguation." *Proceedings of EMNLP 2002*, pp. 25–32.

Frank, E., Hall, M., and Pfahringer, B. (2003). "Locally Weighted Naive Bayes." *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2003, pp. 249–256

Hoste, V., Hendrickx, I., Daelemans, W., and van den Bosch, A. (2002). "Parameter optimization for machine-learning of word sense disambiguation." *Natural Language Engineering*, *8* (3), pp. 311–325.

Ide, N., Véronis, J. (1998). "Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art." *Computational Linguistics 24*, pp. 1–40.

Kilgarriff, A., and Rosenzweig, J. (2000). "Framework and results for English SENSEVAL." *Computers and the Humanities 36*, pp. 15–48.

Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. (1998). "On combining classifiers." *IEEE Transactions on Pattern Analysis and Machine Intelligence 20* (3), pp. 226–239.

Klein, D., Toutanova, K., Tolga Ilhan, H., Kamvar, S. D., and Manning, C. D. (2002). "Combining heterogeneous classifiers for Word-Sense Disambiguation." *ACL WSD Workshop*, pp. 74–80.

Le, C. A., and Shimazu, A. (2004). "High Word Sense Disambiguation using Naive Bayesian classifier with rich features." *The 18th Pacific Asian Conference on Linguistic Information and Computation (PACLIC18)*, pp. 105–113

Le, C. A., Huynh, V. N., and Shimazu, A. (2005). "Combining Classifiers with Multi-Representation of Context in Word Sense Disambiguation." *The 19th Pacific Asian Conference on Knowledge and Data Mining (PAKDD05)*, pp. 262–268.

Leacock, C., Chodorow, M., and Miller, G., (1998). "Using corpus statistics and WordNet relations for Sense Identification." *Computational Linguistics*, pp. 147–165.

Mooney, R. J. (1996). "Comparative experiments on Disambiguating Word Senses: An illustration of the role of bias in machine learning." *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 82–91.

Ng, H. T., and Lee, H. B. (1996). "Integrating multiple knowledge sources to Disambiguate Word Sense: An exemplar-based approach." *Proceedings of the 34th Annual Meeting of the Society for Computational Linguistics (ACL)*, pp. 40–47.

Pedersen, T. (2000). "A simple approach to building ensembles of Naive Bayesian classifiers for Word Sense Disambiguation." *Proceedings of the North American Chapter of the*

Association for Computational Linguistics (NAACL), pp. 63–69.

Perrone, M. P., and Cooper, L. N. (1993). “When networks disagree: Ensemble methods for hybrid neural networks.” *Neural Networks for Speech and Image Processing*, pp. 126–142.

Wang, X. J., and Matsumoto, Y. (2004). “Trajectory based word sense disambiguation.” *Proceedings of the 20th International Conference on Computational Linguistics*, pp. 903–909.