Word Translation Disambiguation Using Bilingual Bootstrapping

Hang Li*
Microsoft Research Asia

Cong Li*
Microsoft Research Asia

This article proposes a new method for word translation disambiguation, one that uses a machine-learning technique called bilingual bootstrapping. In learning to disambiguate words to be translated, bilingual bootstrapping makes use of a small amount of classified data and a large amount of unclassified data in both the source and the target languages. It repeatedly constructs classifiers in the two languages in parallel and boosts the performance of the classifiers by classifying unclassified data in the two languages and by exchanging information regarding classified data between the two languages. Experimental results indicate that word translation disambiguation based on bilingual bootstrapping consistently and significantly outperforms existing methods that are based on monolingual bootstrapping.

1. Introduction

We address here the problem of word translation disambiguation. If, for example, we were to attempt to translate the English noun *plant*, which could refer either to a type of factory or to a form of flora (i.e., in Chinese, either to 工厂 [*gongchang*] or to 植物 [*zhiwu*]), our goal would be to determine the correct Chinese translation. That is, word translation disambiguation is essentially a special case of word sense disambiguation (in the above example, *gongchang* would correspond to the sense of *factory* and *zhiwu* to the sense of *flora*).[1]

We could view word translation disambiguation as a problem of classification. To perform the task, we could employ a supervised learning method, but since to do so would require human labeling of data, which would be expensive, bootstrapping would be a better choice.

Yarowsky (1995) has proposed a bootstrapping method for word sense disambiguation. When applied to translation from English to Chinese, his method starts learning with a small number of English sentences that contain ambiguous English words and that are labeled with correct Chinese translations of those words. It then uses these classified sentences as training data to create a classifier (e.g., a decision list), which it uses to classify unclassified sentences containing the same ambiguous words. The output of this process is then used as additional training data. It also adopts the one-sense-per-discourse heuristic (Gale, Church, and Yarowsky 1992b) in classifying unclassified sentences. By repeating the above process, an accurate classifier for word translation disambiguation can be created. Because this method uses data in a single language (i.e., the source language in translation), we refer to it here as monolingual bootstrapping (MB).

* 5F Sigma Center, No. 49 Zhichun Road, Haidian, Beijing, China, 100080. E-mail:{hangli,i-congl}@microsoft.com.

1 In this article, we take English-Chinese translation as an example; but the ideas and methods described here can be applied to any pair of languages.

In this paper, we propose a new method of bootstrapping, one that we refer to as bilingual bootstrapping (BB). Instead of using data in one language, BB uses data in two languages. In translation from English to Chinese, for example, BB makes use of unclassified data from both languages. It also uses a small number of classified data in English and, optionally, a small number of classified data in Chinese. The data in the two languages should be from the same domain but are not required to be exactly in parallel.

BB constructs classifiers for English-to-Chinese translation disambiguation by repeating the following two steps: (1) Construct a classifier for each of the languages on the basis of classified data *in both languages*, and (2) use the constructed classifier for each language to classify unclassified data, which are then added to the classified data of the language. We can use classified data in both languages in step (1), because words in one language have translations in the other, and we can transform data from one language into the other.

We have experimentally evaluated the performance of BB in word translation disambiguation, and all of our results indicate that BB consistently and significantly outperforms MB. The higher performance of BB can be attributed to its effective use of the asymmetric relationship between the ambiguous words in the two languages.

Our study is organized as follows. In Section 2, we describe related work. Specifically, we formalize the problem of word translation disambiguation as that of classification based on statistical learning. As examples, we describe two such methods: one using decision lists and the other using naive Bayes. We also explain the Yarowsky disambiguation method, which is based on Monolingual Bootstrapping. In Section 3, we describe bilingual bootstrapping, comparing BB with MB, and discussing the relationship between BB and co-training. In Section 4, we describe our experimental results, and finally, in Section 5, we give some concluding remarks.

2. Related Work

2.1 Word Translation Disambiguation

Word translation disambiguation (in general, word sense disambiguation) can be viewed as a problem of classification and can be addressed by employing various *supervised* learning methods. For example, with such a learning method, an English sentence containing an ambiguous English word corresponds to an instance, and the Chinese translation of the word in the context (i.e., the word sense) corresponds to a classification decision (a label).

Many methods for word sense disambiguation based on supervised learning technique have been proposed. They include those using naive Bayes (Gale, Church, and Yarowsky 1992a), decision lists (Yarowsky 1994), nearest neighbor (Ng and Lee 1996), transformation-based learning (Mangu and Brill 1997), neural networks (Towell and Voorhees 1998), Winnow (Golding and Roth 1999), boosting (Escudero, Marquez, and Rigau 2000), and naive Bayesian ensemble (Pedersen 2000). The assumption behind these methods is that it is nearly always possible to determine the sense of an ambiguous word by referring to its context, and thus all of the methods build a classifier (i.e., a classification program) using features representing context information (e.g., surrounding context words). For other related work on translation disambiguation, see Brown et al. (1991), Bruce and Weibe (1994), Dagan and Itai (1994), Lin (1997), Pedersen and Bruce (1997), Schutze (1998), Kikui (1999), Mihalcea and Moldovan (1999), Koehn and Knight (2000), and Zhou, Ding, and Huang (2001).

Let us formulate the problem of word sense (translation) disambiguation as follows. Let E denote a set of words. Let ε denote an *ambiguous* word in E, and let e

denote a *context* word in E. (Throughout this article, we use Greek letters to represent ambiguous words and italic letters to represent context words.) Let T_ε denote the set of senses of ε, and let t_ε denote a sense in T_ε. Let $\mathbf{e}_\varepsilon$ stand for an instance representing a context of ε, that is, a sequence of context words surrounding ε:

$$\mathbf{e}_\varepsilon = (e_{\varepsilon,1}, e_{\varepsilon,2}, \ldots, (\varepsilon), \ldots, e_{\varepsilon,m}), e_{\varepsilon,i} \in E,\ (i = 1, \ldots, m)$$

For the example presented earlier, we have $\varepsilon = \text{plant}, T_\varepsilon = \{1, 2\}$, where 1 represents the sense *factory* and 2 the sense *flora*. From the phrase "... computer manufacturing plant and adjacent..." we obtain $\mathbf{e}_\varepsilon$ = (... computer, manufacturing, (plant), and, adjacent, ...).

For a specific ε, we define a *binary* classifier for resolving each of its ambiguities in T_ε in a general form as[2]

$$P(t_\varepsilon \mid \mathbf{e}_\varepsilon),\ t_\varepsilon \in T_\varepsilon \text{ and } P(\bar{t}_\varepsilon \mid \mathbf{e}_\varepsilon),\ \bar{t}_\varepsilon = T_\varepsilon - \{t_\varepsilon\}$$

where $\mathbf{e}_\varepsilon$ denotes an instance representing a context of ε. All of the supervised learning methods mentioned previously can automatically create such a classifier. To construct classifiers using supervised methods, we need classified data such as those in Figure 1.

2.2 Decision Lists

Let us first consider the use of decision lists, as proposed in Yarowsky (1994). Let f_ε denote a feature of the context of ε. A feature can be, for example, a word's occurrence immediately to the left of ε. We define many such features. For each feature f_ε, we use the classified data to calculate the posterior probability ratio of each sense t_ε with respect to the feature as

$$\lambda(t_\varepsilon \mid f_\varepsilon) = \frac{P(t_\varepsilon \mid f_\varepsilon)}{P(\bar{t}_\varepsilon \mid f_\varepsilon)}$$

For each feature f_ε, we create a rule consisting of the feature, the sense

$$\arg\max_{t_\varepsilon \in T_\varepsilon} \lambda(t_\varepsilon \mid f_\varepsilon)$$

and the score

$$\max_{t_\varepsilon \in T_\varepsilon} \lambda(t_\varepsilon \mid f_\varepsilon)$$

We sort the rules in descending order with respect to their scores, provided that the scores of the rules are larger than the default

$$\max_{t_\varepsilon \in T_\varepsilon} \frac{P(t_\varepsilon)}{P(\bar{t}_\varepsilon)}$$

The sorted rules form an if-then-else type of rule sequence, that is, a decision list.[3] For a new instance $\mathbf{e}_\varepsilon$, we use the decision list to determine its sense. The rule in the list whose feature is *first* satisfied in the context of $\mathbf{e}_\varepsilon$ is applied in sense disambiguation.

2 In this article we always employ binary classifiers even there are multiple classes.
3 We note that there are two types of decision lists. One is defined as here; the other is defined as a conditional distribution over a partition of the feature space (cf. Li and Yamanishi 2002).

P1 . . . Nissan car and truck *plant*. . . (1)
P2 . . . computer manufacturing *plant* and adjacent. . . (1)
P3 . . . automated manufacturing *plant* in Fremont. . . (1)
P4 . . . divide life into *plant* and animal kingdom. . . (2)
P5 . . . thousands of *plant* and animal species. . . (2)
P6 . . . zonal distribution of *plant* life. . . (2)
. . .
. . .

Figure 1
Examples of classified data (ε = plant).

2.3 Naive Bayesian Ensemble

Let us next consider the use of naive Bayesian classifiers. Given an instance $\mathbf{e}_\varepsilon$, we can calculate

$$\lambda^*(\mathbf{e}_\varepsilon) = \max_{t_\varepsilon \in T_\varepsilon} \frac{P(t_\varepsilon \mid \mathbf{e}_\varepsilon)}{P(\bar{t}_\varepsilon \mid \mathbf{e}_\varepsilon)} = \max_{t_\varepsilon \in T_\varepsilon} \frac{P(t_\varepsilon)P(\mathbf{e}_\varepsilon \mid t_\varepsilon)}{P(\bar{t}_\varepsilon)P(\mathbf{e}_\varepsilon \mid \bar{t}_\varepsilon)} \tag{1}$$

according to Bayes' rule and select the sense

$$t^*(\mathbf{e}_\varepsilon) = \arg\max_{t_\varepsilon \in T_\varepsilon} \frac{P(t_\varepsilon)P(\mathbf{e}_\varepsilon \mid t_\varepsilon)}{P(\bar{t}_\varepsilon)P(\mathbf{e}_\varepsilon \mid \bar{t}_\varepsilon)} \tag{2}$$

In a naive Bayesian classifier, we assume that the words in $\mathbf{e}_\varepsilon$ with a fixed t_ε are independently generated from $P(e_\varepsilon \mid t_\varepsilon)$ and calculate

$$P(\mathbf{e}_\varepsilon \mid t_\varepsilon) = \prod_{i=1}^{m} P(e_{\varepsilon,i} \mid t_\varepsilon)$$

Here $P(e_\varepsilon \mid t_\varepsilon)$ represents the conditional probability of e in the context of ε given t_ε. We calculate $P(\mathbf{e}_\varepsilon \mid \bar{t}_\varepsilon)$ similarly. We can then calculate (1) and (2) with the obtained $P(\mathbf{e}_\varepsilon \mid t_\varepsilon)$ and $P(\mathbf{e}_\varepsilon \mid \bar{t}_\varepsilon)$.

The naive Bayesian ensemble method for word sense disambiguation, as proposed in Pedersen (2000), employs a linear combination of several naive Bayesian classifiers constructed on the basis of a number of nested surrounding contexts[4]

$$P(t_\varepsilon \mid \mathbf{e}_\varepsilon) = \frac{1}{h} \sum_{i=1}^{h} P(t_\varepsilon \mid \mathbf{e}'_{\varepsilon,i})$$

$$\mathbf{e}'_{\varepsilon,1} \subset \cdots \subset \mathbf{e}'_{\varepsilon,i} \cdots \subset \mathbf{e}'_{\varepsilon,h} = \mathbf{e}'_\varepsilon \ (i = 1, \ldots, h)$$

The naive Bayesian ensemble is reported to perform the best for word sense disambiguation with respect to a benchmark data set (Pedersen 2000).

2.4 Monolingual Bootstrapping

Since data preparation for supervised learning is expensive, it is desirable to develop bootstrapping methods. Yarowsky (1995) proposed such a method for word sense disambiguation, which we refer to as monolingual bootstrapping.

4 Here $\mathbf{u} \subset \mathbf{v}$ denotes that $\mathbf{u}$ is a sub-sequence of $\mathbf{v}$.

Let L_ε denote a set of classified instances (labeled data) in English, each representing one context of ε:

$$L_\varepsilon = \{(\mathbf{e}_{\varepsilon,1}, t_{\varepsilon,1}), (\mathbf{e}_{\varepsilon,2}, t_{\varepsilon,2}), \ldots, (\mathbf{e}_{\varepsilon,k}, t_{\varepsilon,k})\}$$
$$t_{\varepsilon,i} \in T_\varepsilon \ (i = 1, 2, \ldots, k)$$

and U_ε a set of unclassified instances (unlabeled data) in English, each representing one context of ε:

$$U_\varepsilon = \{\mathbf{e}_{\varepsilon,1}, \mathbf{e}_{\varepsilon,2}, \ldots, \mathbf{e}_{\varepsilon,l}\}$$

The instances in Figure 1 can be considered examples of L_ε. Furthermore, we have

$$L_E = \bigcup_{\varepsilon \in E} L_\varepsilon, U_E = \bigcup_{\varepsilon \in E} U_\varepsilon, T = \bigcup_{\varepsilon \in E} T_\varepsilon,$$

An algorithm for monolingual bootstrapping is presented in Figure 2. For a better comparison with bilingual bootstrapping, we have extended the method so that it

Input: E, T, L_E, U_E, Parameter: b, θ
Repeat the following processes until unable to continue

1. 1 **for each** ($\varepsilon \in E$) {
2 **for each** ($t \in T_\varepsilon$) {
3 use L_ε to create classifier:

$$P(t \mid \mathbf{e}_\varepsilon),\ t \in T_\varepsilon \text{ and } P(\bar{t} \mid \mathbf{e}_\varepsilon),\ \bar{t} \in T_\varepsilon - \{t\};\}\}$$

2. 4 **for each** ($\varepsilon \in E$) {
5 $NU \leftarrow \{\}; NL \leftarrow \{\};$
6 **for each** ($t \in T_\varepsilon$) {
7 $S_t \leftarrow \{\};$
8 $Q_t \leftarrow \{\};\}$
9 **for each** ($\mathbf{e}_\varepsilon \in U_\varepsilon$){
10 calculate $\lambda^*(\mathbf{e}_\varepsilon) = \max_{t \in T_\varepsilon} \frac{P(t \mid \mathbf{e}_\varepsilon)}{P(\bar{t} \mid \mathbf{e}_\varepsilon)};$
11 let $t^*(\mathbf{e}_\varepsilon) = \arg\max_{t \in T_\varepsilon} \frac{P(t \mid \mathbf{e}_\varepsilon)}{P(\bar{t} \mid \mathbf{e}_\varepsilon)};$
12 **if** ($\lambda^*(\mathbf{e}_\varepsilon) > \theta$ & $t^*(\mathbf{e}_\varepsilon) = t$)
13 put $\mathbf{e}_\varepsilon$ into $S_t;\}$
14 **for each** ($t \in T_\varepsilon$){
15 sort $\mathbf{e}_\varepsilon \in S_t$ in descending order of $\lambda^*(\mathbf{e}_\varepsilon)$ and put the top b elements into $Q_t;\}$
16 **for each** ($\mathbf{e}_\varepsilon \in \bigcup_t Q_t$){
17 put $\mathbf{e}_\varepsilon$ into NU and put $(\mathbf{e}_\varepsilon, t^*(\mathbf{e}_\varepsilon))$ into $NL;\}$
18 $L_\varepsilon \leftarrow L_\varepsilon \bigcup NL;$
19 $U_\varepsilon \leftarrow U_\varepsilon - NU;\}$

Figure 2
Monolingual bootstrapping.

performs disambiguation for all the words in E. Note that we can employ any kind of classifier here.

At step 1, for each ambiguous word ε we create binary classifiers for resolving its ambiguities (cf. lines 1–3 of Figure 2). At step 2, we use the classifiers for each word ε to select some unclassified instances from U_ε, classify them, and add them to L_ε (cf. lines 4–19). We repeat the process until all the data are classified.

Lines 9–13 show that for each unclassified instance $\mathbf{e}_\varepsilon$, we classify it as having sense t if t's posterior odds are the largest among the possible senses and are larger than a threshold θ. For each class t, we store the classified instances in S_t. Lines 14–15 show that for each class t, we only choose the top b classified instances in terms of the posterior odds. For each class t, we store the selected top b classified instances in Q_t. Lines 16–17 show that we create the classified instances by combining the instances with their classification labels.

After line 17, we can employ the one-sense-per-discourse heuristic to further classify unclassified data, as proposed in Yarowsky (1995). This heuristic is based on the observation that when an ambiguous word appears in the same text several times, its tokens usually refer to the same sense. In the bootstrapping process, for each newly classified instance, we automatically assign its class label to those unclassified instances that also contain the same ambiguous word and co-occur with it in the same text.

Hereafter, we will refer to this method as monolingual bootstrapping with one sense per discourse. This method can be viewed as a special case of co-training (Blum and Mitchell 1998).

2.5 Co-training

Monolingual bootstrapping augmented with the one-sense-per-discourse heuristic can be viewed as a special case of co-training, as proposed by Blum and Mitchell (1998) (see also Collins and Singer 1999; Nigam et al. 2000; and Nigam and Ghani 2000). Co-training conducts two bootstrapping processes in parallel and makes them collaborate with each other. More specifically, co-training begins with a small number of classified data and a large number of unclassified data. It trains two classifiers from the classified data, uses each of the two classifiers to classify some unclassified data, makes the two classifiers exchange their classified data, and repeats the process.

3. Bilingual Bootstrapping

3.1 Basic Algorithm

Bilingual bootstrapping makes use of a small amount of classified data and a large amount of unclassified data in both the source and the target languages in translation. It repeatedly constructs classifiers in the two languages in parallel and boosts the performance of the classifiers by classifying data in each of the languages and by exchanging information regarding the classified data between the two languages.

Figures 3 and 4 illustrate the process of bilingual bootstrapping. Figure 5 shows the translation relationship among the ambiguous words *plant, zhiwu*, and *gongchang*. There is a classifier for *plant* in English. There are also two classifiers, one each for *zhiwu* and *gongchang*, respectively, in Chinese. Sentences containing *plant* in English and sentences containing *zhiwu* and *gongchang* in Chinese are used.

In the beginning, sentences P1 and P4 on the English side are assigned labels 1 and 2, respectively (Figure 3). On the Chinese side, sentences G1 and G3 are assigned labels 1 and 3, respectively, and sentences Z1 and Z3 are assigned labels 2 and 4, respectively. The four labels here correspond to the four links in Figure 5. For example, label 1 represents the sense *factory* and label 2 represents the sense *flora*. Other sentences are

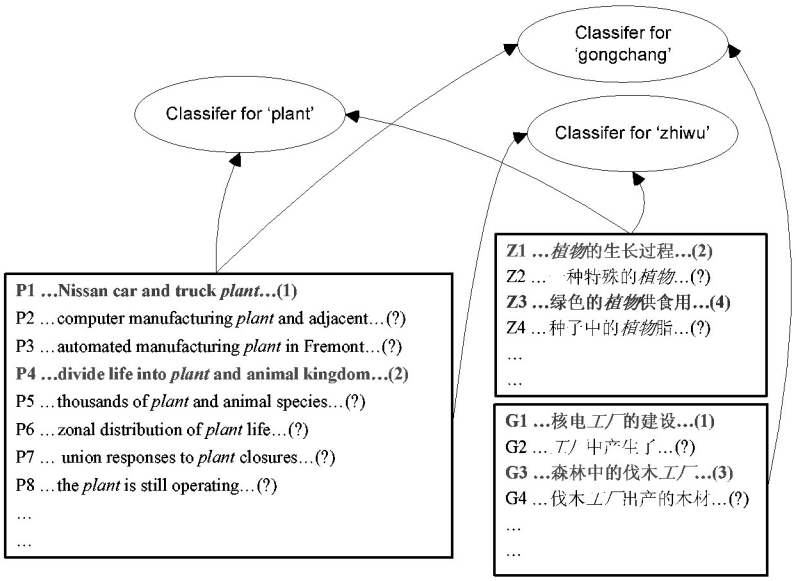

Figure 3
Bilingual bootstrapping (1).

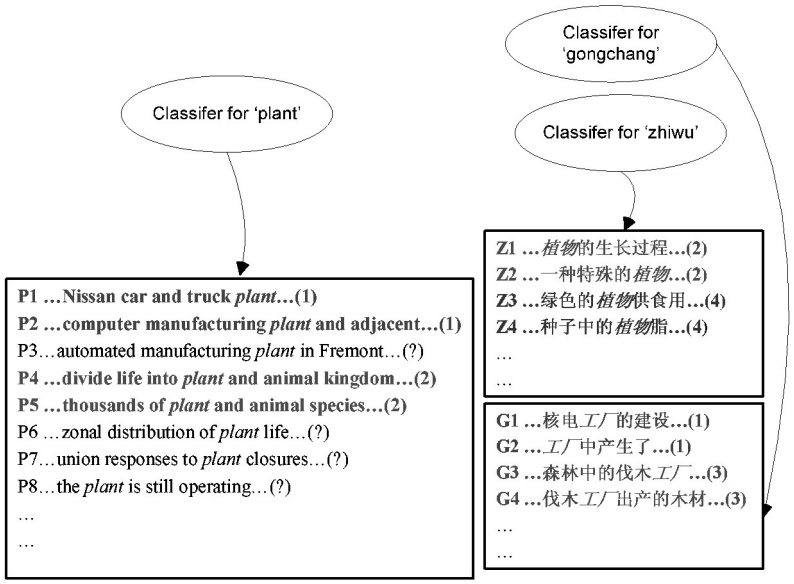

Figure 4
Bilingual bootstrapping (2).

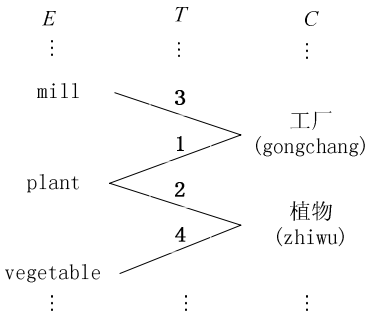

Figure 5
Example of translation dictionary.

not labeled. Bilingual bootstrapping uses labeled sentences P1, P4, G1, and Z1 to create a classifier for *plant* disambiguation (between label 1 and label 2). It also uses labeled sentences Z1, Z3, and P4 to create a classifier for *zhiwu* and uses labeled sentences G1, G3, and P1 to create a classifier for *gongzhang*. Bilingual bootstrapping next uses the classifier for *plant* to label sentences P2 and P5 (Figure 4). It uses the classifier for *zhiwu* to label sentences Z2 and Z4, and uses the classifier for *gongchang* to label sentences G2 and G4. The process is repeated until we cannot continue.

To describe this process formally, let E denote a set of words in English, C a set of words in Chinese, and T a set of senses (links) in a translation dictionary as shown in Figure 5. (Any two linked words can be translations of each other.) Mathematically, T is defined as a *relation* between E and C, that is, $T \subseteq E \times C$. Let ε stand for an ambiguous word in E, and γ an ambiguous word in C. Also let e stand for a context word in E, c a context word in C, and t a sense in T.

For an English word ε, $T_\varepsilon = \{t \mid t = (\varepsilon, \gamma'),\ t \in T\}$ represents the set of ε's possible senses (i.e., its links), and $C_\varepsilon = \{\gamma' \mid (\varepsilon, \gamma') \in T\}$ represents the Chinese words that can be translations of ε (i.e., Chinese words to which ε is linked). Similarly, for a Chinese word γ, let $T_\gamma = \{t \mid t = (\varepsilon', \gamma), t \in T\}$ and $E_\gamma = \{\varepsilon' \mid (\varepsilon', \gamma) \in T\}$.

For the example in Figure 5, when $\varepsilon = plant$, we have $T_\varepsilon = \{1, 2\}$ and $C_\varepsilon = \{gongchang, zhiwu\}$. When $\gamma = gongchang$, $T_\gamma = \{1, 3\}$ and $E_\gamma = \{plant, mill\}$. When $\gamma = zhiwu$, $T_\gamma = \{2, 4\}$ and $E_\gamma = \{plant, vegetable\}$. Note that *gongchang* and *zhiwu* share the senses $\{1, 2\}$ with *plant*.

Let $\mathbf{e}_\varepsilon$ denote an instance (a sequence of context words surrounding ε) in English:

$$\mathbf{e}_\varepsilon = (e_{\varepsilon,1}, e_{\varepsilon,2}, \ldots, e_{\varepsilon,m}),\ e_{\varepsilon,i} \in E\ (i = 1, 2, \ldots, m)$$

Let $\mathbf{c}_\gamma$ denote an instance (a sequence of context words surrounding γ) in Chinese:

$$\mathbf{c}_\gamma = (c_{\gamma,1}, c_{\gamma,2}, \ldots, c_{\gamma,n},\ c_{\gamma,i} \in C\ (i = 1, 2, \ldots, n)$$

For an English word ε, a *binary* classifier for resolving each of the ambiguities in T_ε is defined as

$$P(t_\varepsilon \mid \mathbf{e}_\varepsilon),\ t_\varepsilon \in T_\varepsilon \text{ and } P(\bar{t}_\varepsilon \mid \mathbf{e}_\varepsilon),\ \bar{t}_\varepsilon = T_\varepsilon - \{t_\varepsilon\}$$

Similarly, for a Chinese word γ, a binary classifier is defined as

$$P(t_\gamma \mid \mathbf{c}_\gamma),\ t_\gamma \in T_\gamma \text{ and } P(\bar{t}_\gamma \mid \mathbf{c}_\gamma),\ \bar{t} = T_\gamma - \{t_\gamma\}$$

Let L_ε denote a set of classified instances in English, each representing one context of ε:

$$L_\varepsilon = \{(\mathbf{e}_{\varepsilon,1}, t_{\varepsilon,1}), (\mathbf{e}_{\varepsilon,2}, t_{\varepsilon,2}), \ldots, (\mathbf{e}_{\varepsilon,k}, t_{\varepsilon,k})\}, t_{\varepsilon,i} \in T_\varepsilon\ (i = 1, 2, \ldots, k)$$

and U_ε a set of unclassified instances in English, each representing one context of ε:

$$U_\varepsilon = \{\mathbf{e}_{\varepsilon,1}, \mathbf{e}_{\varepsilon,2}, \ldots, \mathbf{e}_{\varepsilon,l}\}$$

Similarly, we denote the sets of classified and unclassified instances with respect to γ in Chinese as L_γ and U_γ, respectively. Furthermore, we have

$$L_E = \bigcup_{\varepsilon \in E} L_\varepsilon, L_C = \bigcup_{\gamma \in C} L_\gamma, U_E = \bigcup_{\varepsilon \in E} U_\varepsilon, U_C = \bigcup_{\gamma \in C} U_\gamma$$

We also have

$$T = \bigcup_{\varepsilon \in E} T_\varepsilon = \bigcup_{\gamma \in C} T_\gamma$$

Sentences P1 and P4 in Figure 3 are examples of L_ε. Sentences Z1, Z3 and G1, G3 are examples of L_γ.

We perform bilingual bootstrapping as described in Figure 6. Note that we can, in principle, employ any kind of classifier here.

The figure explains the process for English (left-hand side); the process for Chinese (right-hand side) behaves similarly. At step 1, for each ambiguous word ε, we create binary classifiers for resolving its ambiguities (cf. lines 1–3). The main point here is that we use classified data from both languages to construct classifiers, as we describe in Section 3.2. For the example in Figure 3, we use both L_ε (sentences P1 and P4) and L_γ, $\gamma \in C_\varepsilon$ (sentences Z1 and G1) to construct a classifier resolving ambiguities in $T_\varepsilon = \{1, 2\}$. Note that not only P1 and P4, but also Z1 and G1, are related to $\{1, 2\}$. At step 2, for each word ε, we use its classifiers to select some unclassified instances from U_ε, classify them, and add them to L_ε (cf. lines 4–19). We repeat the process until we cannot continue.

Lines 9–13 show that for each unclassified instance $\mathbf{e}_\varepsilon$, we use the classifiers to classify it into the class (sense) t if t's posterior odds are the largest among the possible classes and are larger than a threshold θ. For each class t, we store the classified instances in S_t. Lines 14–15 show that for each class t, we choose only the top b classified instances (in terms of the posterior odds), which are then stored in Q_t. Lines 16–17 show that we create the classified instances by combining the instances with their classification labels. We note that after line 17 we can also employ the one-sense-per-discourse heuristic.

3.2 An Implementation

Although we can in principle employ any kind of classifier in BB, we use here naive Bayes (or naive Bayesian ensemble). We also use the EM algorithm in **classified data transformation** between languages. As will be made clear, this implementation of BB can naturally combine the features of naive Bayes (or naive Bayesian ensemble) and the features of EM. Hereafter, when we refer to BB, we mean this implementation of BB.

We explain the process for English (left-hand side of Figure 6); the process for Chinese (right-hand side of figure) behaves similarly. At step 1 in BB, we construct a naive Bayesian classifier as described in Figure 7. At step 2, for each instance $\mathbf{e}_\varepsilon$, we use the classifier to calculate

$$\lambda^*(\mathbf{e}_\varepsilon) = \max_{t_\varepsilon \in T_\varepsilon} \frac{P(t_\varepsilon \mid \mathbf{e}_\varepsilon)}{P(\bar{t}_\varepsilon \mid \mathbf{e}_\varepsilon)} = \max_{t_\varepsilon \in T_\varepsilon} \frac{P(t_\varepsilon) P(\mathbf{e}_\varepsilon \mid t_\varepsilon)}{P(\bar{t}_\varepsilon) P(\mathbf{e}_\varepsilon \mid \bar{t}_\varepsilon)}$$

Input: $E, C, T, L_E, U_E, L_C, U_C$, Parameter: b, θ

Repeat *in parallel* the following processes for English (left) and Chinese (right), until unable to continue :

1. 1 **for each** ($\varepsilon \in E$) {
2 **for each** ($t \in T_\varepsilon$) {
3 use L_ε and L_γ ($\gamma \in C_\varepsilon$) to create classifier:
$P(t \mid \mathbf{e}_\varepsilon)$, $t \in T_\varepsilon$ & $P(\bar{t} \mid \mathbf{e}_\varepsilon)$, $t = T_\varepsilon - \{t\}$ }}

2. 4 **for each** ($\varepsilon \in E$) {
5 $NU \leftarrow \{\}; NL \leftarrow \{\};$
6 **for each** ($t \in T_\varepsilon$) {
7 $S_t \leftarrow \{\};$
8 $Q_t \leftarrow \{\};$}
9 **for each** ($\mathbf{e} \in U_\varepsilon$){
10 calculate $\lambda^*(\mathbf{e}) = \max_{t \in T_\varepsilon} \frac{P(t \mid \mathbf{e}_\varepsilon)}{P(\bar{t} \mid \mathbf{e}_\varepsilon)}$;
11 let $t^*(\mathbf{e}) = \arg\max_{t \in T_\varepsilon} \frac{P(t \mid \mathbf{e}_\varepsilon)}{P(\bar{t} \mid \mathbf{e}_\varepsilon)}$;
12 **if** ($\lambda^*(\mathbf{e}) > \theta$ & $t^*(\mathbf{e}) = t$)
13 put $\mathbf{e}$ into S_t;}
14 **for each** ($t \in T_\varepsilon$){
15 sort $\mathbf{e} \in S_t$ in descending order of $\lambda^*(\mathbf{e})$ and put the top b elements into Q_t;}
16 **for each** ($\mathbf{e} \in \bigcup_t Q_t$){
17 put $\mathbf{e}$ into NU and put $(\mathbf{e}, t^*(\mathbf{e}))$ into NL;}
18 $L_\varepsilon \leftarrow L_\varepsilon \cup NL;$
19 $U_\varepsilon \leftarrow U_\varepsilon - NU$;}

for each ($\gamma \in C$) {
for each ($t \in T_\gamma$) {
use L_γ and L_ε ($\varepsilon \in E_\gamma$) to create classifier:
$P(t \mid \mathbf{c}_\gamma)$, $t \in T_\gamma$ & $P(\bar{t} \mid \mathbf{c}_\gamma)$, $\bar{t} = T_\gamma - \{t\}$;}}
for each ($\gamma \in C$) {
$NU \leftarrow \{\}; NL \leftarrow \{\};$
for each ($t \in T_\gamma$) {
$S_t \leftarrow \{\};$
$Q_t \leftarrow \{\};$}
for each ($\mathbf{c} \in U_\gamma$){
calculate $\lambda^*(\mathbf{c}) = \max_{t \in T_\gamma} \frac{P(t \mid \mathbf{c}_\gamma)}{P(\bar{t} \mid \mathbf{c}_\gamma)}$;
let $t^*(\mathbf{c}) = \arg\max_{t \in T_\gamma} \frac{P(t \mid \mathbf{c}_\gamma)}{P(\bar{t} \mid \mathbf{c}_\gamma)}$;
if ($\lambda^*(\mathbf{c}) > \theta$ & $t^*(\mathbf{c}) = t$)
put $\mathbf{c}$ into S_t;}
for each ($t \in T_\gamma$){
sort $\mathbf{c} \in S_t$ in descending order of $\lambda^*(\mathbf{c})$ and put the top b elements into Q_t;}
for each ($\mathbf{c} \in \bigcup_t Q_t$){
put $\mathbf{c}$ into NU and put $(\mathbf{c}, t^*(\mathbf{c}))$ into NL;}
$L_\gamma \leftarrow L_\gamma \cup NL;$
$U_\gamma \leftarrow U_\gamma - NU$;}

Output: classifiers in English and Chinese

Figure 6
Bilingual bootstrapping.

We estimate

$$P(\mathbf{e}_\varepsilon \mid t_\varepsilon) = \prod_{i=1}^{m} P(e_{\varepsilon,i} \mid t_\varepsilon)$$

We estimate $P(\mathbf{e}_\varepsilon \mid \bar{t}_\varepsilon)$ similarly. We estimate $P(e_\varepsilon \mid t_\varepsilon)$ by linearly combining $P^{(E)}(e_\varepsilon \mid t_\varepsilon)$ estimated from English and $P^{(C)}(e_\varepsilon \mid t_\varepsilon)$ estimated from Chinese:

$$P(e_\varepsilon \mid t_\varepsilon) = (1 - \alpha - \beta)P^{(E)}(e_\varepsilon \mid t_\varepsilon) + \alpha P^{(C)}(e_\varepsilon \mid t_\varepsilon) + \beta P^{(U)}(e_\varepsilon) \quad (3)$$

where $0 \le \alpha \le 1$, $0 \le \beta \le 1$, $\alpha + \beta \le 1$, and $P^{(U)}(e_\varepsilon)$ is a uniform distribution over E, which is used for avoiding zero probability. In this way, we estimate $P(e_\varepsilon \mid t_\varepsilon)$ using information from not only English, but also Chinese.

We estimate $P^{(E)}(e_\varepsilon \mid t_\varepsilon)$ with maximum-likelihood estimation (MLE) using L_ε as data. The estimation of $P^{(C)}(e_\varepsilon \mid t_\varepsilon)$ proceeds as follows.

For the sake of readability, we rewrite $P^{(C)}(e_\varepsilon \mid t_\varepsilon)$ as $P(e \mid t)$. We define a finite-mixture model of the form $P(c \mid t) = \sum_{e \in E} P(c \mid e, t)P(e \mid t)$, and for a specific ε we assume that the data in

$$L_\gamma = \{(\mathbf{c}_{\gamma,1}, t_{\gamma,1}), (\mathbf{c}_{\gamma,2}, t_{\gamma,2}), \ldots, (\mathbf{c}_{\gamma,h}, t_{\gamma,h})\}, t_{\gamma,i} \in T_\gamma \ (i = 1, \ldots, h), \quad \forall \gamma \in C_\varepsilon$$

estimate $P^{(E)}(e_\varepsilon \mid t_\varepsilon)$ with MLE using L_ε as data;
estimate $P^{(C)}(e_\varepsilon \mid t_\varepsilon)$ with EM algorithm using L_γ for each $\gamma \in C_\varepsilon$ as data;
calculate $P(e_\varepsilon \mid t_\varepsilon)$ as a linear combination of $P^{(E)}(e_\varepsilon \mid t_\varepsilon)$ and $P^{(C)}(e_\varepsilon \mid t_\varepsilon)$;
estimate $P(t_\varepsilon)$ with MLE using L_ε;
calculate $P(e_\varepsilon \mid \bar{t}_\varepsilon)$ and $P(\bar{t}_\varepsilon)$ similarly.

Figure 7
Creating a naive Bayesian classifier.

are generated independently from the model. We can therefore employ the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) to estimate the parameters of the model, including $P(e \mid t)$. Note that e and c represent *context words*.

Recall that E is a set of words in English, C is a set of words in Chinese, and T is a set of senses. For a specific English word e, $C_e = \{c' \mid (e, c') \in T\}$ represents the Chinese words that are its possible translations.

Initially, we set

$$P(c \mid e, t) = \begin{cases} \frac{1}{|C_e|}, & \text{if } c \in C_e \\ 0, & \text{if } c \notin C_e \end{cases}$$

$$P(e \mid t) = \frac{1}{|E|},\ e \in E$$

We next estimate the parameters by iteratively updating them, as described in Figure 8, until they converge. Here $f(c, t)$ stands for the frequency of c in the instances which have sense t. The context information in Chinese $f(c, t_\varepsilon)$ is then "transformed" into the English version $P^{(C)}(e_\varepsilon \mid t_\varepsilon)$ through the links in T.

Figure 9 shows an example of estimating $P(e_\varepsilon \mid t_\varepsilon)$ with respect to the *factory* sense (i.e., sense 1). We first use sentences such as P1 in Figure 3 to estimate $P^{(E)}(e_\varepsilon \mid t_\varepsilon)$ with MLE as described above. We next use sentences such as G1 to estimate $P^{(C)}(e_\varepsilon \mid t_\varepsilon)$ as described above. Specifically, with the frequency data $f(c, t_\varepsilon)$ and EM we can estimate $P^{(C)}(e_\varepsilon \mid t_\varepsilon)$. Finally, we linearly combine $P^{(E)}(e_\varepsilon \mid t_\varepsilon)$ and $P^{(C)}(e_\varepsilon \mid t_\varepsilon)$ to obtain $P(e_\varepsilon \mid t_\varepsilon)$.

3.3 Comparison of BB and MB

We note that monolingual bootstrapping is a special case of bilingual bootstrapping (consider the situation in which $\alpha = 0$ in formula (3)).

BB can always perform better than MB. The asymmetric relationship between the ambiguous words in the two languages stands out as the key to the higher performance

E-step: $P(e \mid c, t) \leftarrow \frac{P(c \mid e, t)P(e \mid t)}{\sum_{e \in E} P(c \mid e, t)P(e \mid t)}$

M-step: $P(c \mid e, t) \leftarrow \frac{f(c, t)P(e \mid c, t)}{\sum_{c \in C} f(c, t)P(e \mid c, t)}$

$$P(e \mid t) \leftarrow \frac{\sum_{c \in C} f(c, t)P(e \mid c, t)}{\sum_{c \in C} f(c, t)}$$

Figure 8
The EM algorithm.

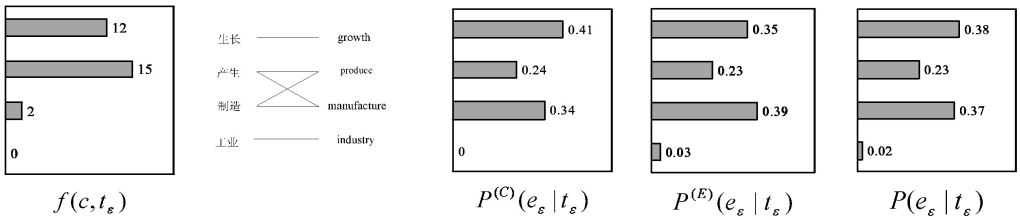

Figure 9
Parameter estimation.

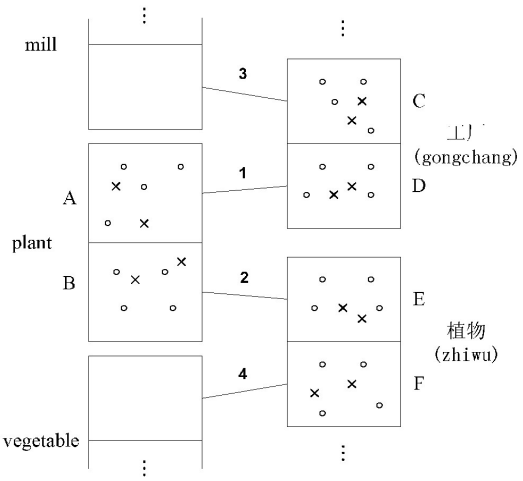

Figure 10
Example application of BB.

of BB. By *asymmetric relationship* we mean the many-to-many mapping relationship between the words in the two languages, as shown in Figure 10.

Suppose that the classifier with respect to *plant* has two classes (denoted as A and B in Figure 10). Further suppose that the classifiers with respect to *gongchang* and *zhiwu* in Chinese each have two classes (C and D) and (E and F), respectively. A and D are equivalent to one another (i.e., they represent the same sense), and so are B and E.

Assume that instances are classified after several iterations of BB as depicted in Figure 10. Here, circles denote the instances that are correctly classified and crosses denote the instances that are incorrectly classified.

Since A and D are equivalent to one another, we can transform the instances with D and use them to boost the performance of classification to A, because the misclassified instances (crosses) with D are those mistakenly classified from C, and they will not have much negative effect on classification to A, even though the translation from Chinese into English can introduce some noise. Similar explanations can be given for other classification decisions.

In contrast, MB uses only the instances in A and B to construct a classifier. When the number of misclassified instances increases (as is inevitable in bootstrapping), its performance will stop improving. This phenomenon has also been observed when MB is applied to other tasks (cf. Banko and Brill 2001; Pierce and Cardie 2001).

3.4 Relationship between BB and Co-training

We note that there are similarities between BB and co-training. Both BB and co-training execute two bootstrapping processes in parallel and make the two processes collaborate with one another in order to improve their performance. The two processes look at different types of information in data and exchange the information in learning. However, there are also significant differences between BB and co-training. In co-training, the two processes use different *features*, whereas in BB, the two processes use different *classes*. In BB, although the features used by the two classifiers are transformed from one language into the other, they belong to the same space. In co-training, on the other hand, the features used by the two classifiers belong to two different spaces.

4. Experimental Results

We have conducted two experiments on English-Chinese translation disambiguation. In this section, we will first describe the experimental settings and then present the results. We will also discuss the results of several follow-on experiments.

4.1 Translation Disambiguation Using BB

Although it is possible to straightforwardly apply the algorithm of BB described in Section 3 to word translation disambiguation, here we use a variant of it better adapted to the task and for fairer comparison with existing technologies. The variant of BB we use has four modifications:

1. It actually employs naive Bayesian ensemble rather than naive Bayes, because naive Bayesian ensemble generally performs better than naive Bayes (Pedersen 2000).
2. It employs the one-sense-per-discourse heuristic. It turns out that in BB with one sense per discourse, there are two layers of bootstrapping. On the top level, bilingual bootstrapping is performed between the two languages, and on the second level, co-training is performed within each language. (Recall that MB with one sense per discourse can be viewed as co-training.)
3. It uses only classified data in English at the beginning. That is to say, it requires exactly the same human labeling efforts as MB does.
4. It individually resolves ambiguities on selected English words such as *plant* and *interest*. (Note that the basic algorithm of BB performs disambiguation on all the words in English and Chinese.) As a result, in the case of *plant*, for example, the classifiers with respect to *gongchang* and *zhiwu* make classification decisions only on D and E and not C and F (in Figure 10), because it is not necessary to make classification decisions on C and F. In particular, it calculates $\lambda^*(\mathbf{c})$ as $\lambda^*(\mathbf{c}) = P(\mathbf{c} \mid t)$ and sets $\theta = 0$ in the right-hand side of step 2.

4.2 Translation Disambiguation Using MB

We consider here two implementations of MB for word translation disambiguation. In the first implementation, in addition to the basic algorithm of MB, we also use (1) naive Bayesian ensemble, (2) one sense per discourse, and (3) a small amount of classified data in English at the beginning. (We will denote this implementation as MB-B hereafter.) The second implementation is different from the first one only in (1). That

Table 1
Data descriptions in Experiment 1.

English words	Chinese words	Senses	Seed words
interest	兴趣	readiness to give attention	show
	利息	money paid for the use of money	rate
	股份, 股权	a share in company or business	hold
	利益	advantage, advancement or favor	conflict
line	绳索, 缆绳	a thin flexible object	cut
	行, 句	written or spoken text	write
	线路	telephone connection	telephone
	队伍, 队列	formation of people or things	wait
	界线, 边界	an artificial division	between
	产品, 品种	product	product

is, it employs a decision list as the classifier. This implementation is exactly the one proposed in Yarowsky (1995). (We will denote it as MB-D hereafter.) MB-B and MB-D can be viewed as the state-of-the-art methods for word translation disambiguation using bootstrapping.

4.3 Experiment 1: WSD Benchmark Data

We first applied BB, MB-B, and MB-D to translation disambiguation on the English words *line* and *interest* using a benchmark data set.[5] The data set consists mainly of articles from the *Wall Street Journal* and is prepared for conducting word sense disambiguation (WSD) on the two words (e.g., Pedersen 2000).

We collected from the HIT dictionary[6] the Chinese words that can be translations of the two English words; these are listed in Table 1. One sense of an English word links to one group of Chinese words. (For the word *interest*, we used only its four major senses, because the remaining two minor senses occur in only 3.3% of the data.)

For each sense, we selected an English word that is strongly associated with the sense according to our own intuition (cf. Table 1). We refer to this word as a **seed word**. For example, for the sense of *money paid for the use of money*, we selected the word *rate*. We viewed the seed word as a classified "sentence," following a similar proposal in Yarowsky (1995). In this way, for each sense we had a classified instance in English. As unclassified data in English, we collected sentences in news articles from a Web site (www.news.com), and as unclassified data in Chinese, we collected sentences in news articles from another Web site (news.cn.tom.com). Note that we need to use only the sentences containing the words in Table 1. We observed that the distribution of the senses in the unclassified data was balanced. As test data, we used the entire benchmark data set.

Table 2 shows the sizes of the data sets. Note that there are in general more unclassified sentences (and texts) in Chinese than in English, because one English word usually can link to several Chinese words (cf. Figure 5).

As the translation dictionary, we used the HIT dictionary, which contains about 76,000 Chinese words, 60,000 English words, and 118,000 senses (links). We then used the data to conduct translation disambiguation with BB, MB-B, and MB-D, as described in Sections 4.1 and Section 4.2.

5 http://www.d.umn.edu/~tpederse/data.html.
6 This dictionary was created by Harbin Institute of Technology.

Table 2
Data set sizes in Experiment 1.

	Unclassified sentences (texts)		
Words	English	Chinese	Test sentences
interest	1,927 (1,072)	8,811 (2,704)	2,291
line	3,666 (1,570)	5,398 (2,894)	4,148

For both BB and MB-B, we used an ensemble of five naive Bayesian classifiers with window sizes of $\pm 1, \pm 3, \pm 5, \pm 7$, and ± 9 words, and we set the parameters β, b, and θ to 0.2, 15, and 1.5, respectively. The parameters were tuned on the basis of our preliminary experimental results on MB-B; they were not tuned, however, for BB. We set the BB-specific parameter α to 0.4, which meant that we weighted information from English and Chinese equally.

Table 3 shows the translation disambiguation accuracies of the three methods as well as that of a baseline method in which we always choose the most frequent sense. Figures 11 and 12 show the learning curves of MB-D, MB-B, and BB. Figure 13 shows the accuracies of BB with different α values. From the results, we see that BB *consistently* and *significantly* outperforms both MB-D and MB-B. The results from the sign test are statistically significant (p-value < 0.001). (For the sign test method, see, for example, Yang and Liu [1999]).

Table 4 shows the results achieved by some existing *supervised* learning methods with respect to the benchmark data (cf. Pedersen 2000). Although BB is a method nearly equivalent to one based on unsupervised learning, it still performs favorably when compared with the supervised methods (note that since the experimental settings are different, the results cannot be directly compared).

4.4 Experiment 2: Yarowsky's Words

We also conducted translation on seven of the twelve English words studied in Yarowsky (1995). Table 5 lists the words we used.

Table 3
Accuracies of disambiguation in Experiment 1.

Words	Major (%)	MB-D (%)	MB-B (%)	BB (%)
interest	54.6	54.7	69.3	**75.5**
line	53.5	55.6	54.1	**62.7**

Table 4
Accuracies of supervised methods.

	interest (%)	line (%)
Naive Bayesian ensemble	89	88
Naive Bayes	74	72
Decision tree	78	—
Neural network	—	76
Nearest neighbor	87	—

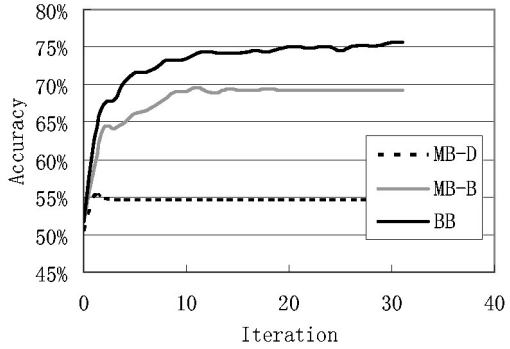

Figure 11
Learning curves with *interest*.

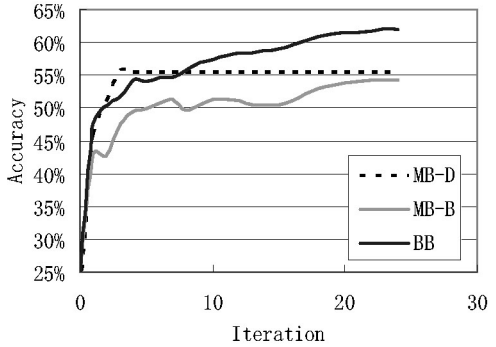

Figure 12
Learning curves with *line*.

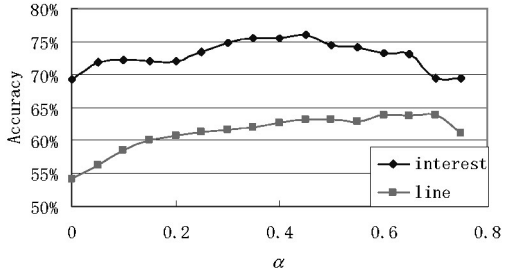

Figure 13
Accuracies of BB with different α values.

Table 5
Data set descriptions in Experiment 2.

English words	Chinese words	Seed words
bass	鱼, 鱼类 / 低音, 低音部	fish / music
drug	药物, 药品 / 毒品	treatment / smuggler
duty	责任, 职责 / 税, 税收	discharge / export
palm	棕榈树, 棕榈 / 手掌	tree / hand
plant	工厂, 厂 / 植物	industry / life
space	空间, 间隙 / 太空, 宇宙空间	volume / outer
tank	坦克 / 水箱, 油箱	combat / fuel

Table 6
Data set sizes in Experiment 2.

	Unclassified sentences (texts)		Test
Words	English	Chinese	sentences
bass	142 (106)	8,811 (4,407)	200
drug	3,053 (1,048)	5,398 (3,143)	197
duty	1,428 (875)	4,338 (2,714)	197
palm	366 (267)	465 (382)	197
plant	7,542 (2,919)	24,977 (13,211)	197
space	3,897(1,494)	14,178 (8,779)	197
tank	417 (245)	1,400 (683)	199
Total	16,845 (6,954)	59,567 (33,319)	1,384

For each of the English words, we extracted about 200 sentences containing the word from the Encarta[7] English corpus and hand-labeled those sentences using our own Chinese translations. We used the labeled sentences as test data and the unlabeled sentences as unclassified data in English. Table 6 shows the data set sizes. We also used the sentences in the Great Encyclopedia[8] Chinese corpus as unclassified data in Chinese. We defined, for each sense, a seed word in English as a classified instance in English (cf. Table 5). We did not, however, conduct translation disambiguation on the words *crane*, *sake*, *poach*, *axes*, and *motion*, because the first four words do not frequently occur in the Encarta corpus, and the accuracy of choosing the major translation for the last word already exceeds 98%.

We next applied BB, MB-B, and MB-D to word translation disambiguation. The parameter settings were the same as those in Experiment 1. Table 7 shows the disambiguation accuracies, and Figures 14–20 show the learning curves for the seven words.

From the results, we see again that BB significantly outperforms MB-D and MB-B. Note that the results of MB-D here cannot be directly compared with those in Yarowsky (1995), because the data used are different. Naive Bayesian ensemble did not perform well on the word *duty*, causing the accuracies of both MB-B and BB to deteriorate.

7 http://encarta.msn.com/default.asp.
8 http://www.whlib.ac.cn/sjk/bkqs.htm.

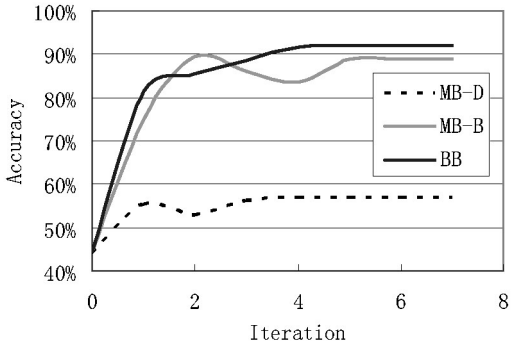

Figure 14
Learning curves with *bass*.

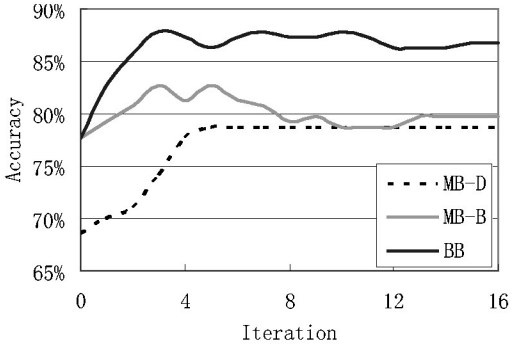

Figure 15
Learning curves with *drug*.

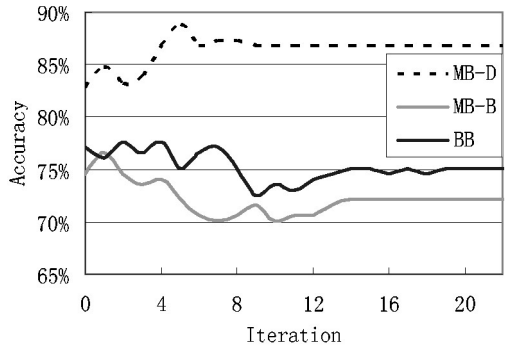

Figure 16
Learning curves with *duty*.

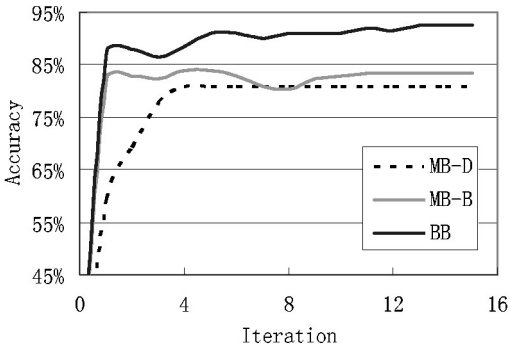

Figure 17
Learning curves with *palm*.

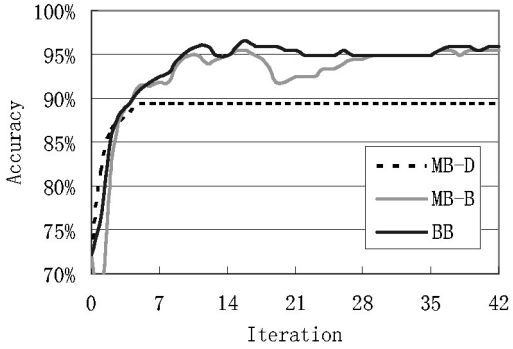

Figure 18
Learning curves with *plant*.

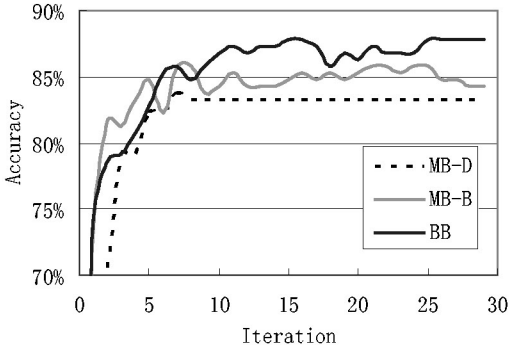

Figure 19
Learning curves with *space*.

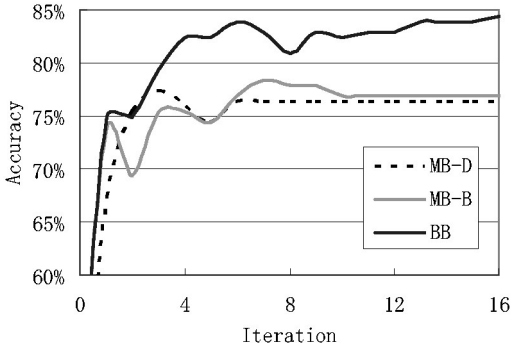

Figure 20
Learning curves with *tank*.

Table 7
Accuracies of disambiguation in Experiment 2.

Words	Major (%)	MB-D (%)	MB-B (%)	BB (%)
bass	61.0	57.0	89.0	**92.0**
drug	77.7	78.7	79.7	**86.8**
duty	86.3	**86.8**	72.0	75.1
palm	82.2	80.7	83.3	**92.4**
plant	71.6	89.3	95.4	**95.9**
space	64.5	83.3	84.3	**87.8**
tank	60.3	76.4	76.9	**84.4**
Total	71.9	78.8	82.9	**87.8**

Table 8
Top words for *interest rate* sense of *interest*.

MB-B	BB
payment	saving
cut	payment
earn	benchmark
short	whose
short-term	base
yield	prefer
u.s.	fixed
margin	debt
benchmark	annual
regard	dividend

4.5 Discussion

We investigated the reason for BB's outperforming MB and found that the explanation in Section 3.3 appears to be valid according to the following observations.

1. In a naive Bayesian classifier, words with large values of likelihood ratio $\frac{P(e|t)}{P(e|\bar{t})}$ will have strong influences on classification. We collected the words having the largest likelihood ratio with respect to each sense t in both BB and MB-B and found that BB obviously has more "relevant words" than MB-B. Here words relevant to a particular sense refer to the words that are strongly indicative of that sense according to human judgments.

Table 8 shows the top 10 words in terms of likelihood ratio with respect to the *interest rate* sense in both BB and MB-B. The relevant words are italicized. Figure 21 shows the numbers of relevant words with respect to the four senses of *interest* in BB and MB-B.

2. From Figure 13, we see that the performance of BB remains high or gets higher even when α becomes larger than 0.4 (recall that β was fixed at 0.2). This result strongly indicates that the information from Chinese has positive effects.

3. One might argue that the higher performance of BB can be attributed to the larger amount of unclassified data it uses, and thus if we increase the amount of unclassified data for MB, it is likely that MB can perform as well as BB. We conducted an additional experiment and found that this is not the case. Figure 22 shows the accuracies achieved by MB-B as the amount of unclassified data increases. The plot shows that the accuracy of MB-B does not improve when the amount of unclassified

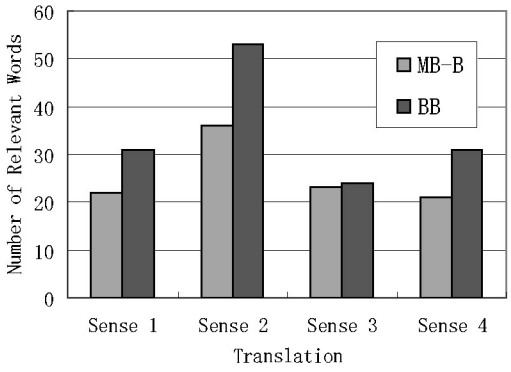

Figure 21
Number of relevant words.

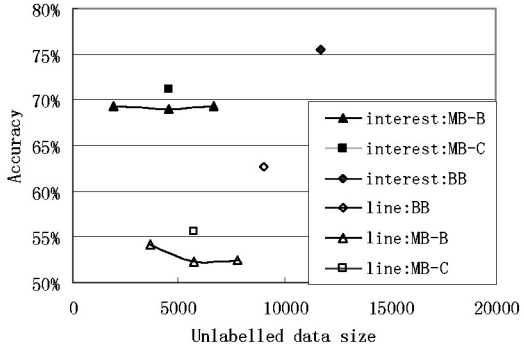

Figure 22
When more unclassified data available.

data increases. Figure 22 plots again the results of BB as well as those of a method referred to as MB-C. In MB-C, we linearly combined two MB-B classifiers constructed with two different unclassified data sets, and we found that although the accuracies are improved in MB-C, they are still much lower than those of BB.

4. We have noticed that a key to BB's performance is the asymmetric relationship between the classes in the two languages. Therefore, we tested the performance of MB and BB when the classes in the two languages are symmetric (i.e., one-to-one mapping).

We performed two experiments on text classification in which the categories were finance and industry, and finance and trade, respectively. We collected Chinese texts from the *People's Daily* in 1998 that had already been assigned class labels. We used half of them as unclassified training data in Chinese and the remaining as test data in Chinese. We also collected English texts from the *Wall Street Journal*. We used them as unlabeled training data in English. We used the class names (i.e., *finance, industry*, and *trade*, as seed data (classified data)). Table 9 shows the accuracies of text classification. From the results we see that when the classes are symmetric, BB cannot outperform MB.

5. We also investigated the effect of the one-sense-per-discourse heuristic. Table 10 shows the performance of MB and BB on the word *interest* with and without the heuristic. We see that with the heuristic, the performance of both MB and BB is improved. Even without the heuristic, BB still performs better than MB with the heuristic.

Table 9
Accuracy of text classification.

Classes	MB-B (%)	BB (%)
Finance and industry	93.2	92.9
Finance and trade	78.4	78.6

Table 10
Accuracy of disambiguation.

	MB-D (%)	MB-B (%)	BB (%)
With one sense per discourse	54.7	69.3	75.5
Without one sense per discourse	54.6	66.4	71.6

5. Conclusion

We have addressed here the problem of classification across two languages. Specifically we have considered the problem of bootstrapping. We find that when the task is word translation disambiguation between two languages, we can use the asymmetric relationship between the ambiguous words in the two languages to significantly boost the performance of bootstrapping. We refer to this approach as bilingual bootstrapping. We have developed a method for implementing this bootstrapping approach that naturally combines the use of naive Bayes and the EM algorithm. Future work includes a theoretical analysis of bilingual bootstrapping (generalization error of BB, relationship between BB and co-training, etc.) and extensions of bilingual bootstrapping to more complicated machine translation tasks.

Acknowledgments

We thank Ming Zhou, Ashley Chang and Yao Meng for their valuable comments and suggestions on an early draft of this article. We acknowledge the four anonymous reviewers of this article for their valuable comments and criticisms. We thank Michael Holmes, Mark Petersen, Kevin Knight, and Bob Moore for their checking of the English of this article. A previous version of this article appeared in *Proceedings of the Fortieth Annual Meeting of the Association for Computational Linguistics*.

References

Banko, Michele, and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France.

Blum, Avrim, and Tom M. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, Madison, WI.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1991. Word sense disambiguation using statistical methods. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 264–270, University of California, Berkeley.

Bruce, Rebecca, and Janyce Weibe. 1994. Word-sense disambiguation using decomposable models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 139–146, New Mexico State University, Las Cruces.

Collins, Michael, and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, University of Maryland, College Park.

Dagan, Ido, and Alon Itai. 1994. Word sense disambiguation using a second language monolingual corpus. *Computational Linguistics*, 20(4):563–596.

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

Escudero, Gerard, Lluis Marquez, and German Rigau. 2000. Boosting applied to word sense disambiguation. In *Proceedings of the 12th European Conference on Machine Learning*, pages 129–141, Barcelona.

Gale, William, Kenneth Church, and David Yarowsky. 1992a. A method for disambiguating word senses in a large corpus. *Computers and Humanities*, 26:415–439.

Gale, William, Kenneth Church, and David Yarowsky. 1992b. One sense per discourse. In *Proceedings of DARPA Speech and Natural Language Workshop*, pages 233–237, Harriman, NY.

Golding, Andrew R., and Dan Roth. 1999. A Winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34:107–130.

Kikui, Genichiro. 1999. Resolving translation ambiguity using non-parallel bilingual corpora. In *Proceedings of ACL '99 Workshop on Unsupervised Learning in Natural Language Processing*, University of Maryland, College Park.

Koehn, Philipp, and Kevin Knight. 2000. Estimating word translation probabilities from unrelated monolingual corpora using the EM algorithm. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 711–715, Austin, TX.

Li, Hang, and Kenji Yamanishi. 2002. Text classification using ESC-based stochastic decision lists. *Information Processing and Management*, 38:343–361.

Lin, Dekang. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 64–71, Universidad Nacional de Educación a Distancia (UNED), Madrid.

Mangu, Lidia, and Eric Brill. 1997. Automatic rule acquisition for spelling correction. In *Proceedings of the 14th International Conference on Machine Learning*, pages 187–194, Nashville, TN.

Mihalcea, Rada, and Dan I. Moldovan. 1999. A method for word sense disambiguation of unrestricted text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 152–158, University of Maryland, College Park.

Ng, Hwee Tou, and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 40–47, University of California, Santa Cruz.

Nigam, Kamal, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2–3):103–134.

Nigam, Kamal, and Rayid Ghani. 2000. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the 9th International Conference on Information and Knowledge Management*, pages 86–93, McLean, VA.

Pedersen, Ted. 2000. A simple approach to building ensembles of naive Bayesian classifiers for word sense disambiguation. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle.

Pedersen, Ted, and Rebecca Bruce. 1997. Distinguishing word senses in untagged text. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 197–207, Providence, RI.

Pierce, David, and Claire Cardie. 2001. Limitations of co-training for natural language learning from large datasets. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, Carnegie Mellon University, Pittsburgh.

Schutze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

Towell, Geoffrey, and Ellen M. Voorhees. 1998. Disambiguating highly ambiguous words. *Computational Linguistics*, 24(1):125–146.

Yang, Yiming, and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, CA.

Yarowsky, David. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 88–95, New Mexico State University, Las Cruces.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.

Zhou, Ming, Yuan Ding, and Changning Huang. 2001. Improving translation selection with a new translation model trained by independent monolingual corpora. *International Journal of Computational Linguistics and Chinese Language Processing*, 6(1):1–26.