Chinese Classifier Assignment Using SVMs

Hui Guo and Huayan Zhong
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794-4400, USA
{huguo, huayan}@cs.sunysb.edu

Abstract

In Chinese, nouns need numeral classifiers to express quantity. In this paper, we explore the relationship between classifiers and nouns. We extract a set of lexical, syntactic and ontological features and the corresponding noun-classifier pairs from a corpus and then train SVMs to assign classifers to nouns. We analyse which features are most important for this task.

1 Introduction

In English, numbers directly modify count nouns, as in ‘two apples’ and ‘five computers’. Numbers cannot directly modify mass nouns; instead, an embedded noun phrase must be formed, e.g. ‘five slices of bread’. However, in Chinese all nouns need numeral classifiers to express quantity[1]. When translating from English to Chinese, we may need to choose Chinese classifiers to form noun phrases. We can see the difference between the two languages in the following two examples:

两*[liang]* 个*[**ge**]* 苹果*[pingguo] (Chinese)*
two apples (English)

and

五*[wu]* 片*[**pian**]* 面包*[mianbao] (Chinese)*
five slices of bread (English)

Noun classifer combinations appear with high frequency in Chinese. There are more than 500 classifiers although fewer than 200 of them are frequently used. Each classifier can only be used with certain classes of noun. Nouns in a class usually have similar properties. For example, nouns that can be used with the classifier ‘根[gen]’ are: ‘稻草’(straw), ‘筷子’(chopstick), ‘管子’(pipe), etc. All these objects are long and thin. However, sometimes nouns with similar properties are in different classes. For example, ‘牛’(cow), ‘马’(horse) and ‘羊’(lamb) are all livestock, but they associate with different classifiers. This means that classifier assignment is not totally rule-based but partly idiomatic.

In this paper, we explore the relationship between classifiers and nouns. We extract a set of features and the corresponding noun-classifier attachments from a corpus and then train SVMs to assign classifers to nouns. In Section 4 we describe our data set. In Section 5 we describe our experiments. In Section 6 we present our results.

2 Related Work

Many Asian languages (e.g. Chinese, Korean, Japanese and Thai) have numeral classifier systems. Previous work on noun-classifier matching has been done in these languages. (Sornlertlamvanich et al., 1994) present an algorithm for selecting an appropriate classifier for a noun in Thai. The general idea is to extract noun-classifier collocations from a corpus, and output a list of noun-classifier pairs with frequency information. During noun phrase generation, the most frequently co-occurring classifier for a given noun is selected. However, no evaluation is reported for this algorithm.

The algorithm described in (Paik and Bond, 2001) generates Japanese and Korean numeral

[1] Proper nouns and bare noun phrases do not need classifiers.

classifiers using semantic classes from an ontology. The authors assigned classifiers to each of the 2,710 semantic classes in the ontology by hand. During generation, nouns in each semantic class are assigned the associated classifier. The classifier assignment accuracy is 81% for Japanese classifiers and 62% for Korean classifiers. However, the evaluation set contains only 90 noun phrases, which is pretty small. Furthermore, it is hard work to attach classifiers to an ontology by hand, and with this approach it is hard to deal with cases like the cattle example mentioned earlier.

(Paul et al., 2002) present a method for extracting classifier information from a bilingual (Japanese-English) corpus based on phrasal correspondences in the sentential context. Bilingual sentence pairs are compared to find noun-classifier collocations. The evaluation was done by a human. The precision is high (84.2%) but the recall is only about 40% because the algorithm does not give output for half of the nouns.

In contrast to these algorithms, our approach: is based on a large data set; uses machine learning; and does not require the attachment of classifiers to an ontology by hand.

3 Support Vector Machines

Support Vector Machines (SVMs) are a type of classifier first introduced in (Boser et al., 1992). In the last few years SVMs have become an important and active field in machine learning research. The SVM algorithm detects and exploits complex patterns in data.

A binary SVM is a *maximum margin classifier*. Given a set of training data $\{x_1, x_2, ..., x_k\}$, with corresponding labels $y_1, y_2, ..., y_k \in \{+1, -1\}$, a binary SVM divides the input space into two regions at a *decision boundary*, which is a separating hyperplane $\langle w, x\rangle + b = 0$ (Figure 1). The decision boundary should classify all points correctly, that is:

$$y_i(\langle w, x_i\rangle + b) > 0, \forall i$$

Also, the decision boundary should have the maximum separating margin with respect to the two classes. If we rescale w and b to make the closest point(s) to the hyperplane satisfy $|\langle w, x_i\rangle + b| = 1$, then the margin equals $1/||w||$ and the problem can be formulated as:

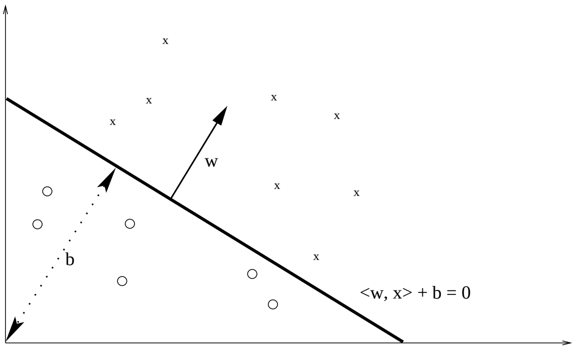

Figure 1: The input space and hyperplane

$$\text{minimize} \quad \frac{1}{2}||w||^2$$

$$\text{subject to} \quad y_i(\langle w, x_i\rangle + b) \geq 1, \forall i$$

The generalized Lagrange Function is:

$$L(w, b, \alpha) = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{l} \alpha_i[y_i(\langle w, x_i\rangle + b) - 1]$$

So we can transform the problem to its dual:
maximize

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{n} \alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle$$

$$\text{subject to} \quad \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

This is a quadratic programming (QP) problem and we can always find the global maximum of α_i. We can recover w and b for the hyperplane by:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$b = -\frac{\max_{y_i=-1}(\langle w, x_i\rangle) + \min_{y_i=+1}(\langle w, x_i\rangle)}{2}$$

If the points in the input space are not linearly separable, we allow ‘slack variables’ ξ_i in the classification. We need to find a soft margin hyperplane, e.g.:

$$\text{minimize} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \quad y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \forall i$$

Once again, a QP solver can be used to find the solution.

For our task we need multi-class SVMs. To get multi-class SVMs, we can construct and combine several binary SVMs (one-against-one), or we can directly consider all data in one optimization formula (one-against-all).

Many SVM implementations are available on the web. We chose LIBSVM (Chang and Lin, 2001), which is an efficient multi-class implementation. LIBSVM uses the “one-against-one” approach in which $k(k-1)/2$ classifiers are constructed and each one trains on data from two different classes (Hsu and Lin, 2002).

4 Data and Resources

We use the Penn Chinese Treebank (Xue et al., 2002) as our corpus and the ontology/lexicon HowNet (Dong and Dong, 2000) to get ontological features for nouns. We train SVMs on different feature sets to see which set(s) of features are important for noun-classifier matching.

4.1 Penn Chinese Treebank

The Penn Chinese Treebank is a 500,000 word Chinese corpus annotated with both part-of-speech (POS) tags and syntactic brackets.

We automatically extract noun phrases that contain classifiers from the corpus. An example noun phrase (translation: ‘a major commercial waterway’) is:

```
(IP
....
 (NP (QP (CD 一) (CLP (M 条)))
   (NP (NN 水运)) (ADJP (JJ 大))
     (NP (NN 动脉)))
...
)
```

The word in (CLP (M 条[tiao])) is the classifier and the head noun of the noun phrase is (NN 动脉). In Section 5.3 we describe a set of features we obtain from each noun phrase and the sentence in which it is embedded.

In our corpus, there are 61587 noun occurrences (12225 unique nouns) and 3940 classifier-noun co-occurrences (212 unique classifiers). However, there is a trival rule determining whether a noun needs a classifier. If a noun is preceded by a quantifier or a determiner, then a classifier is needed, otherwise it is not. Hence, we only focus on noun-classifier pairs. The most frequently occurring classifier in this corpus is ‘个[ge]’, which occurs with 497 unique nouns. In this corpus, 87 classifiers occur in only one noun-classifier pair.

4.2 HowNet

We get ontological features of nouns from HowNet. HowNet is a bilingual Chinese-English lexicon and ontology. Each word sense is assigned to a concept containing ontological features. HowNet uses basic meaning units named sememes to construct concepts.

Table 1 shows an example entry in HowNet. The entry in Table 1 is for the word ‘作家’(writer). The sememe at the first position, ‘human(人)’, is the **categorical attribute**, which describes the general category of the concept. The sememes following the first sememe are **additional attributes**, which give additional specific features. There are two types of pointer, ‘#’ and ‘*’, in the definition. ‘#’ means ‘related’, so ‘#occupation’ shows that the concept has a relationship with ‘occupation’. ‘*’ means ‘agent’, so ‘*compile’ shows that ‘writer’ is the agent of ‘compile’. The sememes ‘#readings’ and ‘literature’ show that the job of ‘writer’ is to compile ‘readings’ about ‘literature’.

We use HowNet 2000, which contains 120,496 entries for about 65,000 Chinese words defined with a set of 1503 sememes. It is big enough for our task and we can get ontological features for 94.71% of the nouns from the Penn Chinese Treebank. For the nouns that are not in HowNet, we just leave the ontological features blank.

5 Experiments

We use six different feature sets to assign classifiers to nouns. To evaluate each feature set, we perform 10-fold cross validation. We report our results in Section 6.

5.1 Baseline Algorithm

In the training data, we count the number of times each classifier appears with a given noun. We assign to each noun in the testing data its most fre-

No.: 114303
W_C (word in Chinese): 作家
E_C (example in Chinese):
G_C (POS tag in Chinese): N
W_E (word in English): writer
E_E (example in English):
G_E (POS tag in English): N
DEF (concept definition): human(人),#occupation(职位),*compile(编辑),
#readings(读物),literature(文)

Table 1: An entry in HowNet

Lexical Features	**Syntactic Features**
noun	POS of noun
first premod	POS of first premod
second premod	POS of second premod
main verb	POS of main verb
total number of premodifiers	sentType
	embedded in vp or pp
	quoted or not

Table 2: Features extracted from training data

quently co-occurring classifier (c.f. (Sornlertlamvanich et al., 1994)). If a noun does not appear in the training data, we assign the classifier ‘个[ge]’, the classifier which appears most frequently overall in the corpus.

5.2 Noun Features

Since classifiers are assigned mostly based on the noun, the most important features for classifier prediction should be features of the nouns. We ran four different experiments for noun features:

- **(1)** The feature set includes only the *noun* itself.
- **(2)** The feature set includes *ontological features* of the noun only. If classifiers are associated with semantic categories (c.f. (Paik and Bond, 2001)), we should be able to assign classifiers based on the ontological features of nouns.
- **(3)** The feature set includes the *noun* and *ontological features*.
- **(4)** Two SVMs are trained: one on the *noun* only, and one on *ontological features* only. During testing, nouns in the training set are assigned classifiers using the first SVM; other nouns are assigned classifiers using the second SVM.

5.3 Context Features

In this set of experiments, we used features from both the noun and the context. The features we used can be categorized into two groups: lexical features and syntactic features. They are shown in Table 2.

We ran two experiments using this set of features:

- **(5)** The feature set includes the noun, lexical and syntactic features only.
- **(6)** The feature set includes the noun, lexical, syntactic and ontological features.

6 Results and Discussion

We built SVMs using all the feature sets described in Section 5 and tested using 10-fold cross validation. We tried the four types of kernel function in LIBSVM: linear, polynomial, radial basis function (RBF) and sigmoid, then selected the RBF kernal $K(x, y) = e^{-\gamma||x-y||^2}$, which gives the

Algorithm	**All nouns**	**Nouns occuring 2+ times**
Baseline	50.76%	50.69%
(1) noun only	57.81% (c = 4, $\gamma = 0.5$)	59.34% (c = 16, $\gamma = 0.125$)
(2) ontology only	58.69% (c = 4, $\gamma = 0.5$)	60.68% (c = 256, $\gamma = 0.125$)
(3) noun **and** ontology	57.81% (c = 16, $\gamma = 0.5$)	59.46% (c = 16, $\gamma = 0.125$)
(4) noun **or** ontology	58.71%	60.55%
(5) noun, syntactic and lexical features	52.14% (c = 1024, $\gamma = 0.5$)	53.51% (c = 16, $\gamma = 0.5$)
(6) all features	52.06% (c = 1024, $\gamma = 0.075$)	53.55% (c = 16, $\gamma = 0.5$)

Table 3: Accuracy of different algorithms

	Most common noun	位[**wei**]	次[**ci**]	个[**ge**]	名[**ming**]	届[**jie**]	项[**xiang**]
位[**wei**]	官员 (official)			24.1 (57.1)	14.7 (34.7)		
次[**ci**]	大会 (convention)			22.3 (53.3)		1.1 (2.6)	7.6 (18.2)
个[**ge**]	项目 (project)	1.0 (7.0)			0.7 (5.2)	0.2 (1.7)	3.3 (24.4)
名[**ming**]	人员 (person)	31.7 (55.2)		23.8 (41.4)			
届[**jie**]	运动会 (sports tournament)	1.9 (2.1)	29.6 (34.0)	31.5 (36.2)			
项[**xiang**]	成果 (achievement)		6.6 (11.3)	35.2 (60.4)		1.1 (1.9)	

Table 4: Most commonly misclassified classifiers; Cell shows percentage of total occurrences of row value misclassified as column value and (percentage of total misclassifications of row value misclassified as column value)

highest accuracy. For each feature set, we systematically varied the values for the parameters C (range from 2^{-5} to 2^{15}) and γ (range from 2^3 to 2^{-15}); we report the best results with corresponding values for C and γ. Finally, for each feature set, we ran once on all nouns and once only on nouns occurring twice or more in the corpus.

Classifier assignment accuracy is reported in Table 3. The performance of all the SVMs is significantly better than baseline (paired t-test, $p < 0.005$). There is no significant difference between the performance with the 1st, 2nd, 3rd and 4th feature sets. But the performance of the SVMs using lexical and syntactic features (experiments 5 and 6) is significantly worse than the performance on feature sets 1-4 (df = 17.426, $p < 0.05$).

These results show that lexical and syntactic contextual features do not have a positive effect on the assignment of classifiers. They confirm the intuition that the noun is the single most important predictor of the classifier; however, the semantic class of the noun works as well as the noun itself. In addition, a combination approach that uses semantic class information when the noun is previously unseen does not perform better.

We also computed the confusion matrix for the most commonly misclassified classifiers. The results are reported in Table 4.

For these experiments we used automatic evaluation (cf. (Paul et al., 2002)). A classifier is only judged to be correct if it is exactly the same as that in the original test set. For some noun phrases, there are multiple valid classifiers. For example, we can say

‘一[yi] 块[**kuai**] 金牌[jinpai]’

or

‘一[yi] 枚[**mei**] 金牌[jinpai]’

(a golden medal).

We did a subjective evaluation on part of our data to evaluate how many automatically generated classifiers are acceptable to human readers. We randomly selected 241 noun-classifier pairs from our data. We presented the sentence containing each pair to a human judge who is a native speaker of Mandarin Chinese. We asked the judge to rate all the classifiers generated by our

Algorithm	Number rated 1 or higher	Percent rated 1 or higher	Average rating
Baseline	209	86.7%	1.59
(1) noun only	224	92.9%	1.76
(2) ontology only	226	93.8%	1.78
(3) noun **and** ontology	226	93.8%	1.77
(4) noun **or** ontology	227	94.2%	1.80
(5) noun, syntactic and lexical features	218	90.5%	1.67
(6) all features	218	90.5%	1.67
Original	241	100%	1.95

Table 5: Human evaluation: Ratings of classifiers

algorithms as well as the original classifier by indicating whether each is good (2), acceptable (1) or bad (0) in that sentence context. The classifiers were presented in random order; the judge was blind to the source of the classifiers.

The results for our human evaluation are reported in Table 5. Although our automatic evaluation indicates relatively poor accuracy, 94.2% of generated classifiers using feature set 4) are rated acceptable or good in our subjective evaluation. Also, the performance of SVMs with the 1st, 2nd, 3rd and 4th feature sets is significantly better than baseline (paired t-test, $p < 0.005$). There is no significant difference between the performance with the 1st, 2nd, 3rd and 4th feature sets. But the performance of the SVMs using lexical and syntactic features (experiments 5 and 6) is significantly worse than those without ($p < 0.05$). The ratings of the classifiers generated by all our algorithms are significantly worse than the original classifiers in the corpus. In future work, we plan to extend this evaluation using more judges.

Which classifier to select also depends on the emotional background of the discourse (Fang, 2003). For example, we can use different classifiers to express different affect for the same noun (e.g. if a government official is in favor or disgrace). However, we cannot get this kind of information from our corpus.

7 Conclusions and Future Work

Our machine learning approach to classifier assignment in Chinese performs better than previously published rule-based approaches and works for bigger data sets. The noun is clearly the most important feature (experiment 1). However, we still think ontological features may be useful in classifier assignment, for example for previously unseen nouns, and our experimental results show a trend in this direction, although not a statistically significant one (experiments 2 and 4).

We used the Chinese Treebank for these experiments because it is the only available corpus of parsed Chinese text. Now that we have isolated the relevant features for this task, we plan to conduct further experiments using larger corpora, such as the Chinese Gigaword (Graf and Chen, 2003).

Our use of ontological features could be improved in several ways. First, the ontological features we get from HowNet do not fit our purpose well. For example, the definitions of ‘猫’ (cat) and ‘牛’ (cow) are both ‘livestock’; however, they should use different classifiers. In order to improve the performance of our approach, we need an ontology that correctly groups nouns into classes according to their semantic properties (e.g. type, shape, color, size).

For another knowledge-rich approach, we could use a complex ontology plus a Chinese classifier dictionary that describes the properties of the objects each classifier can modify. By comparing noun properties and classifier characteristics, classifier assignment could be improved as long as the nouns are in the ontology. However, there are many idiomatic noun-classifier matchings that can not be categorised by dictionaries. Therefore, a combination of rule-based and machine-learning approaches seems most promising.

Third, we can classify Chinese classifers into groups and focus on those that modify single objects. Certain Chinese classifiers can be used before all plural nouns. Some classifiers specify the container of the objects, for example, ‘一[yi] 篮子[**lanzi**] 苹果[pingguo]’ (a basket of apples). The classifier changes when the container changes. These can be treated differently from sortal and anaphoric classifiers.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback, HowNet for giving us permission to download their knowledge base, and Chih-Chung Chang and Chih-Jen Lin for the implementation of LIBSVM.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010.

References

B. Boser, I. Guyon, and V. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory (COLT 1992)*.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Z. Dong and Q. Dong. 2000. Introduction to HowNet - Chinese message structure base. http://www.keenage.com.

L. Fang. 2003. Research of Chinese lexicon teaching - quantities. *Journal of Secondary Education*.

D. Graf and K. Chen. 2003. Chinese gigaword. LDC Catalog Number LDC2003T09.

C. Hsu and C. Lin. 2002. A comparison of methods for multi-class support vector machines. In *IEEE Transactions on Neural Networks*.

K. Paik and F. Bond. 2001. Multilingual generation of numeral classifiers using a common ontology. In *Proceedings of the 19th International Conference on Computer Processing of Oriental Languages*.

M. Paul, E. Sumita, and S. Yamamoto. 2002. Corpus-based generation of numeral classifier using phrase alignment. In *Proceedings of the 19th International Conference on Computational Linguistics*.

V. Sornlertlamvanich, W. Pantachat, and S. Meknavin. 1994. Classifier assignment by corpus-based approach. In *Proceedings of the 15th Conference on Computational Linguistics*.

N. Xue, F. Chiou, and M. Palmer. 2002. Building a large-scale annotated Chinese corpus. In *Proceedings of the 19th International Conference on Computational Linguistics*.