SIR-NERD: A Chinese Named Entity Recognition and Disambiguation System using a Two-Stage Method

Zehuan Peng Le Sun Xianpei Han
Institute of Software, Chinese Academy of Sciences
HaiDian, Beijing, PRC.
{zehuan, sunle, xianpei}@nfs.iscas.ac.cn

Abstract

This paper presents our SIR-NERD system for the Chinese named entity recognition and disambiguation Task in the CIPS-SIGHAN joint conference on Chinese language processing (CLP2012). Our system uses a two-stage method and some key techniques to deal with the named entity recognition and disambiguation (NERD) task. Experimental results on the test data shows that the proposed system, which incorporates classifying and clustering techniques, can achieve competitive performance.

1 Introduction

Named entity recognition and disambiguation (NERD) is an important task in information retrieval (IR) and natural language processing (NLP). Given a set of documents, a NERD system should recognize all named entities within them, and disambiguate them by either linking them to knowledge base entries or grouping names into clusters, with each resulting group a specific entity. Compared with the English NERD, the Chinese NERD has some special challenges: Firstly, many common words can often be used as named entities, too. For example, both the common adjective word "高明(brilliant)" and the common noun "高峰(peak)" are also common male names in China. In these situations, it is challenging to distinguish common words from named entities, and the lack of morphology information in Chinese (such as the Capital word for named entity) further increases the difficulty. Secondly, the Chinese entity name is usually highly ambiguous on entity types, i.e., the same name may refer to many different types of named entities. For example, 金山(Gold Hill) can be used as the name of persons, locations and organizations; 黄河(Yellow River) can be used as name of persons or rivers. Thirdly, it is common that many persons share the same name. For example, the name 李明(Li Ming) or 高峰(Gao Feng) is very popular in China.

In recent years, NERD has attracted a lot of research attention, and most of the research work focus on clustering the observations of a specific name, with each resulting cluster corresponding to a specific entity. Song et al. 2009 proposed a locality-based *tfidf* framework for document representation and similarity measure for webpages clustering. Chen et al. 2007 proposed several token-based and phrase-based features for clustering webpages containing the same person, and achieved a significant improvement of disambiguation performance for web people search.

In the SIR-NERD system, we adopt a two-stage method which can incorporate classifying and clustering techniques for the personal name entity disambiguation task. In the first stage, the system preprocess the corpus through, word segmentation, general named entity recognition, and calculate the similarity between two documents. In the second stage, we group documents into clusters using the agglomerative hierarchical clustering approach, so that each cluster corresponds to a specific entity.

The paper is organized as follows. Section 2 describes the task; Section 3 describes the SIR-NERD system in detail; Section 4 describes the

Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 115–120,
Tianjin, China, 20-21 DEC. 2012

experiments and discusses the results; finally we give a conclusion.

2 Task Description

The named entity recognition and disambiguation task in CIPS-SIGHAN 2012 is a combination of classifying and clustering tasks. There are 16 names in the training data and 32 names in the test data. For each name *N*, there is a document collection *T* and knowledge base (KB) which contains several persons, organizations or locations who share the same name *N*. For each document in *T* (the name *N* in a document is supposed only refer to one entity), the task is to find the target entity of the name *N* in KB; if the target entity of the name *N* in document is not contained in KB, then the system needs to determine whether *N* is a common word or not; if not, we need to cluster these documents into subsets, each of which refers to one single entity. Table 1 shows a KB example for the name 白雪, which contains seven entities. For each entity, a detailed introduction is given.

Id	Introduction
1	*A singer come from Zhejiang* "祖籍浙江省温州市...歌手...浙江军区文工团...歌唱演员..马剑"
2	*A famous actress* "白百何...女演员...白雪...中央戏剧学院...《幸福在哪里》...《与青春有关的日子...《失恋 33 天》...电影"
3	*A woman marathon champion* "女子马拉松冠军得主"
4	*A woman dubber* "女性配音演员...毕业于北京电影学院....黄渤、边江、邱秋、孟宇、张磊、王凯、刘特、褚珺...女性角色"
5	*A famous painter* "陈大威...白雪...河北省涿州市...画家...教授...人民日报社...编委...副院长...北京国际奥林匹克书画院名誉院长..."
6	*A famous after-80s writer* "80 后唯美派和悲情派...作家...雪...吉林，满族人...陕西省安康市，后随父母搬往河南省新乡市"
7	*A heroine in a novel* "孙皓晖...《大秦帝国之黑色裂变》...女主角。白雪...政商白圭之女...智慧胆识..."

Table 1: A KB example for the name 白雪

Table 2 gives three documents containing the name 白雪. If 白雪 in a document refers to an entity in KB, the system should identify its target entity *id* in KB; if 白雪 in a document is a common word with the meaning of "white snow", the system should classify the document into class *other*; if 白雪 refers to an entity not in KB, the system classifies the document into class *out.*

Doc	Content	Target Entity ID
007	"...女子马拉松白雪突破历史..."	3
031	"...天空飘着白雪，四川汶川..."	other
050	"...白雪...《橘子红了》..."	out

Table 2: three typical document examples

3 SIR-NERD system

According to the task requirements, the SIG-NERD system divides the NERD task into two subtasks. Given a document containing name *N*, the first subtask is to classify the document into *id*, *out* or *other*, correspondingly means refering to an entity in KB, an entity not contained in KB and a common word; the second subtask is to cluster documents which are classified as *out* in the first subtask. The two-stage NERD framework of SIR-NERD is illustrated as Figure 1.

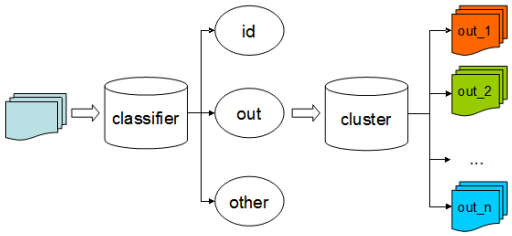

Figure 1: the two-stage NERD framework

In the classification subtask, we first preprocess the corpus through four steps: data clearing, word segmentation and initial named entity recognition, representing documents and entities with selected features, similarity calculation. The steps are described in detail as follows:

- **Data clearing.** In this step, we clear the data by removing XML tags and some unrecognizable characters.

- **Initial NER.** In this step, we use the SIG-NER tool to do the initial word segment and named entity recognition
- **Representing Document**. In this step, we represent each document or entity with some selected features in the context, such as person names, location names, organization names and occupation words
- **Similarity calculation**. In this step, we calculate the similarity between documents and entities based on cosine similarity

In the clustering subtask, we split it into two steps: document representation and hierarchical clustering. The main work is as follows:

- Representing the documents to be clustered with some selected features in the context.
- Using hierarchical clustering method to cluster documents with class label *out*

3.1 Classification

In order to avoid the cascaded error propagation, we determine the class label of a document in one step. For example, in order to process the name 白雪, we use the 7 entities named 白雪 in KB, and treat the *other* and the *out* classes as two pseudo-entities. Each entity is viewed as a class, so 白雪 has 9 classes and now the problem is how to represent these classes. With the document representation, a document containing 白雪 is classified into class with the highest similarity score. As shown above, our SIR-NERD system divides the subtask into four steps: preprocessing, initial NER, documents or entities representation, similarity calculation.

3.1.1 Preprocessing

We are provided with the following data:

- Knowledge base, providing a XML file for each name, the file is named as N.xml, for example 白雪.xml.
- Document collection, for each name N there are a group of xxx.txt files, each of which contain the name N at least once, xxx is a unique document id.
- Answer file, for each name N a answer file with the name N.ans is provided, which records the class label of each document in the document collection.

We use *python xmlparser* to remove all XML tags in XML files and unrecognizable characters in documents.

3.1.2 Initial NER

We use the SIR-NER tool to do the initial named entity recognition. SIR-NER is a Chinese NER tool developed by the SIR laboratory, [1] which does well in general named entity recognition tasks. Taking the following sentence for example:

"*足球运动员，曾效力青岛贝莱特，长春亚泰足球俱乐部队。07 赛季租借到广州医药*"

The NER result is as follows:

"*足球/n 运动员/n ，/w 曾/d 效力/v 青岛/LOC 贝莱特/PER ，/w 长春/LOC 亚泰/nz 足球/n 俱乐部队/n 。/w 07/NUM 赛季/n 租借/v 到/v 广州/LOC 医药/n*"

Named entities like 长春, 青岛 and 广州 can be recognized easily, but for the NRED task in CIPS-SIGHAN 2012, the performance is bad because most names in this task are also common words. For example, SIR-NER system regards the word 白雪 as a common word "snow white" without considering the context. The precision of other words in training data is showed in Table 3.

word	precise	word	precise
丛林	0.0	华山	1.0
方正	0.0	杜鹃	0.0
白云	0.0	雷雨	0.0
高山	0.133	高峰	0.0
高明	0.067	黄河	1.0
...	...	...	...

Table 3: the precision of recognizing the target name as a NE by SIR-NER

3.1.3 Document and entity representation

After the initial NER processing, vector space model is used to represent documents in collection *T* and entities in KB. Different from the traditional *BoW* (bag of words) model, our system use entities to represent the document.

[1] Storage & Information Retrieval, ISCAS. www.icip.org.cn

That is because if we use all words, a lot of noise will be introduced. Experimental results show that using words within the following tags in Table 4 as features achieves encouraging performance.

ORG	A NE, an organization name
LOC	A NE, location name
PER	A NE, a personal name
n	Not a NE, a common noun
vn	Not a NE, a noun-verbs
nz	Not a NE, a proper noun

Table 4: tags used to represent documents and entities

In Table 4, a NE with tag like ORG, LOC and PER contributes 80 percent of the NED precision. The potential reason is that an entity usually semantically related with other entities in the same document.

Furthermore, the occupation description of a person plays an important role in distinguishing different people. For example, a person with the occupation of 教授 *professor* and a person with occupation of 歌手 *singer* tend to be two different people. Therefore, our SIR-NERD system maintains an occupation dictionary, which is built as follows:

- Select 30 occupation words as seeds , such as 总统, 教授, 歌手, 画家, 演员, 局长...
- Use the seeds to expand the occupation dictionary with HIT synonyms dictionary[2] .
- Repeat step two twice, at last we get 1078 occupation words, the new added occupation words are 骑手, 庄园主, 名家, 农民工, 针灸师, 学者, and so on.

In our system, the occupation features are given a higher weight compared with other features when represent documents or entities.

Entities representation

For each name, entities in KB are represented using features with tags in Table 4 and features in the occupation dictionary. Each entity is represented as a vector, in which the features weight with *tfidf* value. *tf* is the times of a word appears in the entity description, *idf* is the number of the entities whose descriptions contain the word.

As described above, we have defined two pseudo entities for each name. The *other* pseudo entity describes the situation that the name is used as a common word and the *out* pseudo entity represents the target entities which are not contained in KB.

In order to represent the *other* pseudo entity, we use nouns which have a high co-occurrence rate with the common word *N*. The co-occurrence rate is calculated as formula (1):

$$co(name, word) = \frac{d(name, word)}{d(name) + d(word)} \quad (1)$$

$d(name, word)$ is the number of documents which contain both *name N* and *word*. $d(name)$ is the number of documents which contain *name*, $d(word)$ is the number of documents which contain *word*. Because the given dataset is not big enough to given a robust co-occurrence estimation, we use the Web as the external source for estimation. The candidate nouns come from two sources: for a name in the training data, document labels are given so we can randomly pick one document with label *other* and use nouns in the document as candidates; also we can search the whole internet with the name as a query, nouns in the top returned documents can be used as candidates. We choose top 20 nouns with high rate. For example for the name 白雪, we get the following list:

"*雪,公主,树,草,山,玉,花,叶,心,光,马,天空,气,人间,大地,生命,微笑,白色,水,心灵,地,深处,太阳,雪花,脚步,月光,光芒,森林,明月,天,灵魂,风景*"

Intuitively, if used as a common word "snow white", 白雪 has a strong semantic association with words like *风景, 公主, 树, 雪花,白色* and so on. So the word lists for 白雪 is reasonable. The weight of each noun in the vector can be computed with the co-occurrence rate.

The representation of the second pseudo entity is also challenging, it describes entities which are not in KB. As discussed above an entity usually has a strong relation with NE like persons, locations and organizations, so when NEs in a document are all not in the NE set in KB, then the document tends to describe an entity not in KB. Based on the hypothesis, we represent the second pseudo entity as follows:

- For each name, we pick out several documents from the doc collections. The

[2]http://ir.hit.edu.cn/phpwebsite/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=162

documents chosen should not contain any NE which appears in KB NE set.

- Select words from the chosen documents with tags in the Table 4 as features, features weight using *tfidf* as above.

Till now we have proposed vector representation methods for three typical entities. Features of different types usually provide different ability for name disambiguation. In order to measure the ability, we define a parameter for each word with tags in Table 4 and each feature in the occupation dictionary. Experiments on the training data show that the weight in Table 5 will result the best performance.

Label	Para name	weight
LOC	v_1	0.715
ORG	v_2	0.429
PER	v_3	0.358
n	v_4	0.191
vn	v_5	0.239
nz	v_6	0.286
occupati -on dict	v_7	1.80

Table 5: parameter values of word labels

Based on the initial weight in Table 5, the weight of the feature words can be calculated as formula (2):

$$w = v_i \times tf \times idf \quad (2)$$

If the words appear in the occupation dictionary the weight can be computed as formula (3):

$$w = v_7 \times v_i \times tf \times idf \quad (3)$$

Document representation

Different from the entity representation in KB, a document is represented using NE words in the document and features in the context instead of using all features in the document. The features should have tags in Table 4, and the weight of each feature is calculated as the same as entity representation.

3.1.3 Similarity Calculation

With the above three steps, we represent each entity as a vector E and each document as a vector D, then the similarity between the two vectors is calculated as formula (4) :

$$sim(e,d) = \frac{\sum_{i=0}^{n} e_i \times d_i}{\sqrt{\sum_{i=0}^{n} e_i^2}\sqrt{\sum_{i=0}^{n} d_i^2}} \quad (4)$$

According to the similarity measure, the document is labeled as the entity label with the highest score.

4 Clustering

Because the number of clusters is not clear, we use agglomerative hierarchical clustering method to divide documents with class label *out* into clusters. Each cluster corresponds to a specific named entity. The algorithm of the bottom-up method is as follows:

1. Treat each document as a single cluster.
2. Calculate the similarity between any two clusters.
3. Merge the two clusters with the highest similarity score into a new cluster.
4. Repeat step 2 and 3 until that any similarity is small than a threshold which is calculated in the training data.

There are three methods to compute similarity between two different clusters: single linkage clustering, group-average linkage clustering and complete linkage clustering. The first step is all the same: calculating the similarity between a document in one cluster and a document in the other cluster. Single linkage clustering uses the largest similarity between data points as clusters similarity; group-average linkage clustering uses the average similarity as clusters similarity; while complete linkage clustering uses the smallest similarity as clusters similarity. In our experiments, we use the group-average linkage methods.

5 Experiment and evaluation

We experiment our system on the training data. The evaluation method is given in the task description in the official website. Precision, recall and F1 value are used as the measurements

to evaluate the system performance. Experiment result on the training data is shown in Table 6:

	precision	recall	F1
白雪	0.8152	0.8670	0.8403
白云	0.6491	0.8112	0.7212
丛林	0.9143	0.8731	0.8932
杜鹃	0.8942	0.8791	0.8866
方正	0.8818	0.8674	0.8745
高超	0.8455	0.9005	0.8721
高峰	0.7937	0.8313	0.8121
高明	0.7795	0.8904	0.8313
高山	0.8804	0.9401	0.9093
高雄	0.8305	0.9401	0.9093
胡琴	0.9623	0.9748	0.9685
华明	0.9716	0.9605	0.9660
华山	0.7721	0.8761	0.8208
黄海	0.7919	0.8426	0.8165
黄河	0.6638	0.8400	0.7416
雷雨	0.8852	0.9263	0.9053
total	0.8332	0.8790	0.8555

Table 5: experiment results on training data

The performance of SIR-NERD system on the test data set is as follows: the precision is 0.7948, the recall is 0.8098 and the F1 value is 0.8022.

6 Conclusion

This paper presents the SIR-NERD system for task 2 in CIPS-SIGHAN 2012. We proposed a two-stage named entity recognition and disambiguation framework, in the first stage we classify the documents into three categories, in the second stage we use the agglomerative hierarchical cluster algorithm to divide the documents with class label *out* into subsets，each resulting cluster corresponds to a specific entity. The key techniques of the SIR-NERD system are:

- We identify that occupation is a discriminant feature for name disambiguation, so we build an occupation dictionary for capturing such features.
- Instead of using all words in a document, we use only entities and occupations for document representation and entity representation, which reduces the noise in representation.

Acknowledgments
The work is supported by the National Natural Science Foundation of China under Grants no. 90920010 and 61100152.

References

Fei song, Robin cohen, Song Lin, Web People Search Based on Locality and Relative Similarity Measures, Proceedings of WWW 2009.

Ying Chen and Martin J.H. CU-COMSEM: Exploring Rich Features for Unsupervised web Personal Name Disambiguation, Proceedings of ACL Semeval 2007.

E. Elmacioglu, Y. F. Tan, S. Yan, M.-Y. Kan, and D. Lee. Psnus: Web people name disambiguation by simple clustering with rich features. In SemEval, 2007.

R. Guha and A. Garg. Disambiguating people in search. In Stanford University, 2004.