Fine-Grained Classification of Named Entities by Fusing Multi-features

Wenjie Li Jiwei Li Ye Tian Zhifang Sui[*]
Key Laboratory of Computational Linguistics, Peking University, Beijing, 100871, China
{lwj, bdlijiwei, ytian, szf}@pku.edu.cn

ABSTRACT

Due to the increase in the number of classes and the decrease in the semantic differences between classes, fine-grained classification of Named Entities is a more difficult task than classic classification of NEs. Using only simple local context features for this fine-grained task cannot yield a good classification performance. This paper proposes a method exploiting Multi-features for fine-grained classification of Named Entities. In addition to adopting the context features, we introduce three new features into our classification model: the cluster-based features, the entity-related features and the class-specific features. We experiment on them separately and also fused with prior ones on the subcategorization of person names. Results show that our method achieves a significant improvement for the fine-grained classification task when the new features are fused with others.

KEYWORDS : Named Entities, fine-grained classification, cluster-based features, entity-related features, class-specific features.

∗ Corresponding author.

Proceedings of COLING 2012: Posters, pages 693–702,
COLING 2012, Mumbai, December 2012.

1 Introduction

The named entity categories defined by the classic Named Entity Classification (NEC) task are coarse grained, typically PERS, LOC, ORG, MISC. The results obtained from coarse grained NEC are insufficient for complex applications such as Information Retrieval, Question-Answering or Ontology Population. Consequently, some researchers turn to address the problem of recognizing and categorizing fine-grained NE classes. Fleischman (2001) presents a preliminary study on the subcategorization of location names, and more recent work focuses on the subcategorization of person names (Fleischman et al., 2002; Giuliano, 2009; Asif Ekbal et al., 2010).

Fine-grained NEC (FG-NEC) is a more difficult task than classic NEC, due to the increase in the number of classes and the decrease in the semantic differences between classes. The classic NEC can yield a good classification performance using only simple local context features. While for the FG-NEC, just using these features is far from enough to meet the requirements.

Take the following sentence for example,

"Dennis Rodman, a close friend of *Pippen*'s who won three NBA Champions with *Jordan*'s Bulls, was shocked to hear of *Pippen*'s comments.",

Based on the context information "NBA Champions", it is easy to recognize "*Pippen*" as an athlete. However for the person "*Dennis Rodman*", using simple context information is difficult to classify it. Therefore FG-NERC needs extended context and semantic features. Acquiring more context information from other related entity mentions in the same text for each entity mention (like "*Pippen*" for "*Dennis Rodman*") and extracting the class-specific feature words (like "NBA Champions" for athlete) may improve the classification results.

In addition, many prior works indicate that the performance of the model just using the lexical features is always limited by the data sparsity. Classic bag-of-words model does not work when there are few matching terms between feature word vectors. For example, there are two context word sets: set1={kitten, nyc} and set2={cat, new, york}. There is no similarity between the terms in each set. Address this limitation, prior works use word clusters from large unannotated corpora as additional features (Ang Sun et al., 2011). These features have been proved to be very useful for alleviating such data sparsity problem. Inspired by this, we also intend to introduce this cluster-based features into our model.

Combining these motivations, we present a method exploiting Multi-features for fine-grained classification of NEs in this paper. The only input data for our algorithm is a few manually annotated entities for each class. In addition to adopting the context word features and the word sense disambiguation features proposed by prior work, this paper puts forward three new features: the cluster-based features, the entity-related features and the class-specific features.

1. *Cluster-based features are generated by the Brown clustering algorithm (Peter F. Brown et al., 1992) from a large unlabeled corpus.*
2. *Entity-related features are context features introduced by other related entities.*
3. *Class-specific features are words extracted for each class. Each word is given a class-specific score denoting its ability to indicate the relevant class.*

Our work presented here concentrates on the subcategorization of person names, since the previous researches have indicated that the classification of person names which relies on much more contextual information are often more challenging. The person instances are already identified as entities, and only being classified into the fine-grained classes here. We choose Maximum Entropy (MaxEnt) model[1] which has already been widely used for a variety of NLP tasks, and proven to be a viable and competitive algorithm in the classification domain. In the following sections, we will describe the proposed features in detail.

2 Features

2.1 Context Features

Context words are the most frequently used features in the prior work. This is based on the assumption that entities occurring in similar contexts belong to the same class. In order to exclude the interference of unrelated words, we extract the words within a window for each entity mention. Only three individual word tokens and their PoS tags before and after the occurrence of the mention will be added into the feature set. In this paper, a context word and its PoS tag are tied together as an ensemble feature. For an entity mention W_i, its context words will be represented as: $f_{i-3}^{i+3}=(w_{i-3}\&pos_{i-3})\cdots(w_{i+3}\&pos_{i+3})$.

2.2 Cluster-based Features

Bag-of-words model cannot deal with synonyms. To address this flaw, some work took advantage of the cluster-based features. The preliminary idea of using word clusters as features was presented by Miller et al. (2004), who augmented name tagging training data with hierarchical word clusters generated by the Brown clustering algorithm (Peter F. Brown et al., 1992) from a large unlabeled corpus.

Ang Sun et al. (2011) use the Brown algorithm to generate the word clusters as additional features which are applying to improve the performance of the relation extraction system. They use the English portion of the TDT5 corpora as their unlabeled data for inducing word clusters. The result of this word clusters is a binary tree. A particular word can be assigned a binary string by following the path from the root to itself in the tree, assigning a 0 for each left branch, and a 1 for each right branch. Each word occupies a leaf in the hierarchy, but each leaf might contain more than one word. The example bit strings of word clusters can be seen from Table 1.

Bit string	Word examples
11111110010111	Poland, Sweden, Australia …
1111001110000	preventing, protecting …
110010011	spokespeople, spokesmen …
110110110001	cup, finals, champions …
1101111101100	senator, citizen …
1101111101110	legislator, lawmaker …

TABLE 1 – Sample bit strings and their corresponding words

[1] In this paper we use the OpenNLP MaxEnt package (http://maxent.sourceforge.net).

In our work, we directly adopt this word clusters result supplied by Ang Sun et al. (2011)[2] to expanding the context features. Without further processing, we exploit the smallest granularity of clusters, just considering the leaf node in the binary tree in our method. For the features extracted in Section 2.1, if a feature word can be found as a leaf in the binary tree, the bit string of this leaf will be added as the additional features into the final feature set.

2.3 Entity-Related Features

The traditional classification methods focus only on the local context features described in Section 2.1. Actually, the local context might not provide sufficient information. In order to improve the performance of fine-grained classification, we want to find more context information.

Gale et al. (1992) state and quantify the observation that words strongly tend to exhibit only one sense in a given discourse or document. Inspired by the view, we discover that in the same passage the person instances appearing together are very likely belong to the same class. We expect to take advantage of this regularity to obtain more contexts for each entity mention.

Looking back to the example mentioned in Section 1, for the person "*Dennis Rodman*", there is no useful local contextual information and we can hardly recognize it as an athlete. However, in that sentence, appearing together with another person "*Pippen*" which can be easily identified as an athlete is a clue that "*Dennis Rodman*" is an athlete too.

Entity-related features are selected based on the assumption that if two entity mentions *A* and *B* often appear together in the same passage, then *A* and *B* are most likely to be the instances of the same class. Before feature sets construction, we can add the local context features of *A* into the feature set of *B*, and vice versa. From such features expanding process, each mention will obtain more sufficient context information. We extract entity-related features as follows:

1. ***Related contexts.*** *For an entity mention A and the text T that contains A, if another mention B appears in T and the distance between A and B is within a length of K, the context features of B which are introduced in Section 2.1 will be added into the feature set of A. In this paper we consider two mentions separated by not more than 10 words are highly related. We set K to 10.*
2. ***Relativity.*** *A binary feature that identifies whether the mentions are related. Since not all entities appear together are actually related, we try to extract the words which always co-occur with instances of the same class, and utilize these words to judge whether multi-mentions appearing together are related. This is based on the fact that when instances of the same class appear in the same text, some words always co-occur with high frequency, e.g. words representing coordination like and or along. For the training corpus, we collect all words co-occur with the same class instances, choosing the top M most frequent words into a word set. Empirically, we set M to 2000 in our work. Given a classification mention A and its related mention B, if their context words hit the word in the word set, this binary feature is set to 1.*

2.4 Class-Specific Features

For the classification task, the feature words representing the semantic information for each class are very important. Similar to the example mentioned in Section 1, the person "*Pippen*" co-

[2] http://www.cs.nyu.edu/~asun/data/TDT5_BrownWC.tar.gz.

occurs in the context with "NBA Champions", we know the proper word "NBA Champions" always co-occurs with the *athlete* instances rather than other class instances, so we regard this proper word as a class-specific feature for class *athlete*.

Therefore, we create the class-specific word sets for each class. The class-specific word set for a class is a list of words, in which each word is given a class-specific score denoting its ability to indicate the relevant class. Each class-specific word set constructs a relevant domain resource for the corresponding class.

Afterwards, we will describe how to choose the class-specific feature word sets for each class. These feature words are derived from the context word features described in Section 2.1. For all unigrams in a window of 3 surrounding the entity mentions in the entire training data, only nouns and verbs are kept as the candidate class-specific feature words. In our work, the same word with different PoS tags will be regarded as different ones. Assuming that there are n classes, namely $C_1 C_2 \cdots C_n$, the class-specific score of the candidate word m for the class C_j is computed as follows.

$$Weight_{C_j}(m) = \frac{Frequency_{c_j}(m)}{\sum_{k=1}^{n} Frequency_{c_k}(m)}$$

$Frequency_{C_j}(m)$ represents the frequency of the word m co-occurring with instances of class C_j; the denominator is the frequency of m co-occurs with all class instances. For the class C_j, only those candidate words of which class-specific scores exceed the threshold t are kept; the retained words constitute the class-specific word set for C_j. In our experiments, the threshold t is empirically set to 0.8.

This weight formula shows that the word occurring with instances of the specific class C_j more times than other class instances will achieve a bigger score. This word represents strong semantic domain information for C_j. We know the domain distribution knowledge is very important for classification. If a mention co-occurs with this word, it would be very likely an instance of C_j.

Class	Word	PoS tag	Weight
Musician	ballet	NNS	1.0
	symphony	NN	0.94
	…	…	…
Poet	ode	NNS	1.0
	sonnet	NN	0.9
	…	…	…
Physicist	mercury	NN	1.0
	equation	NN	0.9
	…	…	…

TABLE 2 – Subset of class-specific feature words generated from training data

After constructing the class-specific word sets (see Table 2), we define a binary feature for each class that checks whether the context of entity mention W_i contains the word in the relevant class-specific word set. If context words surrounding W_i hit the word in the class-specific feature set of C_j, the binary feature corresponding to C_j is set to 1.

3 Experiments

3.1 Experimental Settings

We test our approach on UKWAC[3](M. Baroni et al., 2009), a 2 billion word English corpora constructed from the Web limiting the crawl to the *.uk* domain which has been PoS-tagged and lemmatized. The input person instances for each class are the same as used by Giuliano (2009) based on the People Ontology defined by Giuliano and Gliozzo (2008). The ontology extracted from WordNet is arranged in a multi-level taxonomy with 21 fine-grained classes, containing 1,657 distinct person instances. The taxonomy has a maximum depth of 4.

We extract all entity mentions together with their contexts in the entire corpus. All the contexts in which NEs occur are randomly partitioned into two equally sized subsets. One is used for training and the other for testing, and vice versa. Like other hierarchical classification tasks, the hypernym classes contain all instances of their hyponym classes when constructing the datasets. For example, *Mozart* is an instance of class *Musician* and also regarded as an instance of *Artist*.

The evaluation for hierarchical classification tasks is more complicated. The serious misclassification errors (e.g., an entity mention of class *Musician* is classified as the irrelevant class *Writer*) will be treated differently as the minor errors (e.g., an entity mention of class *Musician* is classified as the super-class *Artist*). In this paper we use the evaluation metric proposed by Melamed and Resnik (2000).

3.2 Experimental Results

We take the model only applying the context features as the baseline, and try to observe the different performance of mixing other features described in Section 2 with the context features. The results are reported in Table 3.

Feature set	Micro-F_1	Macro-F_1
Context Features	50.8	42.1
Context Features & Cluster-based Features	55.2	46.5
Context Features & Entity-related Features	52.4	43.6
Context Features & Class-specific Features	65.2	62.9
All features	79.6	76.5

TABLE 3 – Comparison among the different composite features sets

According to Table 3, the performances of all the composite feature sets are better than the baseline. The baseline using only local context features has the worst performance, achieving an F_1 value of about 50.8%. However, for the coarse grained classification of NEs, currently proposed works (William J. Black et al., 2009) show that using these local context features can achieve an F_1 value of above 80%. In Table 3, the model combining all the features achieves the best performance, a Micro-F_1 of about 79.6%.

Comparison among different features: According to Table 3, the composite feature set applying the class-specific features overperforms the others. Let us review the definition of these

[3] http://wacky.sslmit.unibo.it/doku.php?id=corpora

features. Class-specific features are words that we extract for each class. Each word is given a class-specific score denoting its ability to indicate the relevant class. Actually, these words construct a relevant domain resource for the classification task. Therefore, using these feature words can improve the performance significantly. Since the cluster-based features and entity-related features attempt to expand more information from just the local context window words, the performance of them is not as good as class-specific features. The cluster-based features can expand lexical representation of the feature words. The entity-related features bring in wider contexts through expanding the features from other related entity mentions. For the fine-grained classification task, larger contexts are expected to be employed. For this reason, when the cluster-based features and entity-related features are introduced, their performance is still better than the baseline in Table 3.

Comparison on different levels: Then, we want to evaluate the classification performance on different levels of granularities. According to the People Ontology, the general class person is on the level 1. Table 5 shows the levels which each class belongs to. For each level, both training and test entity mentions belong to the classes from the topmost level to the current level. Table 4 shows the results for different levels. The performance decreases as the level getting lower. Coarser grained classification on higher level has a better performance. For the six classes at level 2, fusing all the features achieves a high Micro-F_1 value of about 92.1%. This indicates that fine grained classification is more difficult.

Level	Context Features		Context Features & Cluster-based Features		Context Features & Entity-related Features		Context Features & Class-specific Features		All features	
	Micro -F_1	Macro -F_1	Micro -F_1	Macro -F_1	Micro -F_1	Macro -F_1	Micro -F_1	Macro -F_1	Micro -F_1	Macro -F_1
1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
2	59.5	48.7	61.2	51.4	61.0	50.7	77.9	73.0	92.1	91.6
3	52.2	41.6	56.3	46.7	55.0	44.9	67.1	64.0	81.6	78.2
4	50.8	42.1	55.2	46.5	52.4	43.6	65.2	62.9	79.6	76.5

TABLE 4 – Comparison of performance of the different composite feature sets on different levels

Comparison to other work: We compare our best performance against the other systems. Fleischman and Hovy (2002) uses the decision trees algorithms and achieve an F_1 value of about 70.4% on held-out data. Claudio Giuliano (2009) classifies person instances into one of the People Ontology classes. They collect more semantic information for the entity instances from the search engines and Wikipedia, achieving an F_1 value of about 80.2%. For the same 21 fine-grained classes, our method classifies each person instance mention in context, while acquire a comparable performance. Asif Ekbal et al. (2010) use an unsupervised pattern-based method to automatically construct a gold standard dataset for this task, the system solely using the context features achieves the F_1 value of 82.6%. They also use UKWAC as their corpus. However, the automatically generated training and test datasets are only based on the appositional patterns, not including all the mentions which can be found in context. These datasets are not representative. Because of different settings and corpus used, the comparison is not convincing. Nevertheless, our experimental results demonstrate that combining these multi-features can achieve a better performance for NEs classification. Table 5 shows the overall view of the best result for each class combining all the features.

Class	Confusion all the features		
	Prec.	Recall	F_1
Creator (2)	63.8	92.1	75.4
Artist (3)	73.9	81.6	77.6
Musician (4)	90.3	57.6	70.3
Painter (4)	92.9	58.5	71.8
Film Maker (3)	89.5	73.1	80.5
Communicator (2)	71.7	86.0	78.2
Representative (3)	97.7	72.9	83.5
Writer (3)	78.9	82.5	80.7
Poet (4)	96.4	58.0	72.4
Dramatist (4)	94.7	57.5	71.6
Scientist (2)	54.3	90.4	67.8
Physicist (3)	87.7	60.0	71.3
Chemist (3)	87.9	58.2	70.0
Social scientist (3)	88.0	59.8	71.2
Mathematician (3)	87.8	59.8	71.1
Biologist (3)	87.3	58.5	70.1
Health professional (2)	84.4	97.7	90.5
Businessperson (2)	89.3	100.0	94.4
Performer (2)	70.2	86.2	77.4
Musician (3)	88.8	74.0	80.7
Actor (3)	88.3	73.4	80.2

TABLE 5 – The results for each class combining all the features (Number *n* in brackets means the corresponding class is arranged in the *n*-th level)

Conclusion and perspectives

This paper presents a method exploiting multi-features for fine-grained classification of Named Entities. We test our approach on UKWAC corpus and classify a candidate entity instance into one of a multi-level taxonomy with 21 fine-grained classes. We experiment on the different composite feature sets and compare the performance on different levels. The results show that these features are useful for this fine-grained classification task.

The remaining problem is that the instance seeds as input should be unambiguous. We need to manually specify them. Though Asif Ekbal et al. (2010) propose a method to automatically construct a dataset, the entity mentions are extracted based only on appositional patterns. The dataset dose not include all the mentions which can be found in context. In order to automatically build training examples for NEs classification, we consider applying more class labels and using these labels to extract the unambiguous entities. This is based on the assumption that ambiguous entity instances for one class always have common labels with other classes.

Acknowledgments

This work is supported by NSFC Project 61075067 and National Key Technology R&D Program (No: 2011BAH10B04-03). We thank Claudio Giuliano for their input person instances and the anonymous reviewers for their insightful comments.

References

Michael Fleischman. (2001). Automated subcategorization of named entities. In *Proceedings of the ACL 2001 Student Workshop*.

Michael Fleischman and Eduard Hovy. (2002). Fine grained classification of named entities. In *Proceedings Of COLING-02*, pages 1–7.

Claudio Giuliano and Alfio Gliozzo. (2008). Instance-based ontology population exploiting named-entity substitution. In *Proceedings of COLING-ACL-08*, pages 265–272.

Claudio Giuliano. (2009). Fine-grained classification of named entities exploiting latent semantic kernels. In *Proceedings of CoNLL-09*, pages 201–209.

Asif Ekbal, Eva Sourjikova, Anette Frank, and Simone Ponzetto. (2010). Assessing the Challenge of Fine-grained Named Entity Recognition and Classification. In *Proceedings of the ACL 2010 Named Entity Workshop (NEWS)*.

Ang Sun , Ralph Grishman , Satoshi Sekine. (2011). Semi-supervised relation extraction with large-scale word clustering. In *Proceedings of ACL-11*, pages 521–529.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, pages 467–479.

Scott Miller, Jethran Guinness and Alex Zamanian. (2004). Name Tagging with Word Clusters and Discriminative Training. In *Proceedings of HLT-NAACL*, pages 337–342.

Gale, W., K. Church, and D. Yarowsky. (1992). A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, pages 415–439.

M. Baroni, S. Bernardini, A. Ferraresi and E. Zanchetta. (2009). The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation*, pages 209–226.

Dan Melamed and Philip Resnik. (2000). Tagger evaluation given hierarchical tag sets. *Computers and the Humanities*, pages 79–84.

William J. Black , Argyrios Vasilakopoulos. (2002). Language independent named entity classification by modified transformation-based learning and by decision tree induction. In *proceeding of CoNLL*, pages1–4.