Focused Named Entity Recognition using Machine Learning

Li Zhang
IBM China Research Laboratory
No. 7, 5th Street, ShangDi
Beijing 100085, P.R.China
lizhang@cn.ibm.com

Yue Pan
IBM China Research Laboratory
No. 7, 5th Street, ShangDi
Beijing 100085, P.R.China
panyue@cn.ibm.com

Tong Zhang
IBM T.J. Watson Research Center
Route 134, Yorktown Heights
NY 10598, U.S. A.
tongz@watson.ibm.com

ABSTRACT

In this paper we study the problem of finding most topical named entities among all entities in a document, which we refer to as focused named entity recognition. We show that these focused named entities are useful for many natural language processing applications, such as document summarization, search result ranking, and entity detection and tracking. We propose a statistical model for focused named entity recognition by converting it into a classification problem. We then study the impact of various linguistic features and compare a number of classification algorithms. From experiments on an annotated Chinese news corpus, we demonstrate that the proposed method can achieve near human-level accuracy.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural language Processing—*Text Analysis*; H.3.1 [**Information Storage And Retrieval**]: Content Analysis and Indexing—*Linguistic processing*

General Terms

Algorithms, Experimentation

Keywords

naive Bayes, decision tree, robust risk minimization, text summarization, topic identification, information retrieval

1. INTRODUCTION

With the rapid growth of online electronic documents, many technologies have been developed to deal with the enormous amount of information, such as automatic summarization, topic detection and tracking, and information retrieval. Among these technologies, a key component is to identify the main topics of a document, where topics can be represented by words, sentences, concepts, and named entities. A number of techniques for this purpose have been proposed in the literature, including methods based on position [3], cue phrases [3], word frequency, lexical chains[1] and discourse segmentation [13]. Although word frequency is the easiest way to representing the topics of a document, it was reported in [12] that position methods produce better results than word counting based methods.

Important sentence extraction is the most popular method studied in the literature. A recent trend in topic sentence extraction is to employ machine learning methods. For example, trainable classifiers have been used in [8, 21, 5, 11] to select sentences based on features such as cue phrase, location, sentence length, word frequency and title, etc.

All of the above methods share the same goal of extracting important sentences from documents. However, for topic representation, sentence-level document summaries may still contain redundant information. For this reason, other representations have also been suggested. For example, in [17], the authors used structural features of technical papers to identify important concepts rather than sentences. The authors of [9] presented an efficient algorithm to choose topic terms for hierarchical summarization according to a probabilistic language model. Another hybrid system, presented in [7], generate summarizations with the help of named entity foci of an article. These named entities include people, organizations, and places, and untyped names.

In this paper, we study the problem of finding important named entities from news articles, which we call *focused named entity recognition*. A news article often reports an event that can be effectively summarized by the *five W* (who, what, when, where, and why) approach. Many of the five W's can be associated with appropriate named entities in the article. Our definition of focused named entities is mainly concerned with Who and What. Therefore it is almost self-evident that the concept of focused named entity is important for document understanding and automatic information extraction. In fact, a number of recent studies have already suggested that named entities are useful for text summarization [15, 4, 7, 16]. Moreover, we shall illustrate that focused named entities can be used in other text processing tasks as well. For example, we can rank search results by giving more weights to focused named entities.

We define focused named entities as named entities that are most relevant to the main topic of a news article. Our

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.
Copyright 2004 ACM 1-58113-881-4/04/0007 ...$5.00.

task is to automatically select these focused named entities from the set of all entities in a document. Since focused named entity recognition is a newly proposed machine learning task, we need to determine whether it is well-posed. That is, whether there exists a sufficient level of agreement on focused named entities among human reviewers. A detailed study on this matter will be reported in the section 5.2. The conclusion of our study is that there is indeed a sufficient level of agreement. Encouraged by this study, we further investigated the machine learning approach to this problem, which is the focus of the paper. We discuss various issues encountered in the process of building a machine learning based system, and show that our method can achieve near human performance.

The remainder of this paper is organized as follows. In Section 2 we introduce the problem of focused named entity recognition and illustrate its applications. Section 3 describes a general machine learning approach to this problem. In Section 4, we present features used in our system. Section 5 presents a study of human-level agreement on focused named entities, and various experiments which illustrate the importance of different features. Some final conclusions will be given in section 6.

2. THE PROBLEM

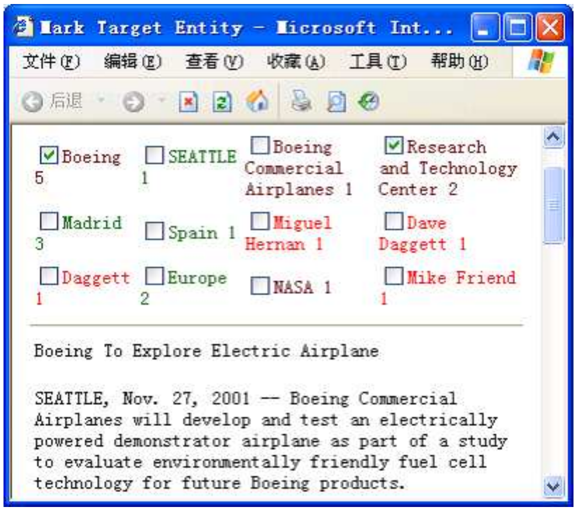
Figure 1 is an example document.[1] This article reports that Boeing Company would work with its new Research and Technology Center to develop a new style of electric airplane. On the upper half of the page, we list all named entities appearing in the article and mark the focused entities. Among the twelve named entities, ”Boeing Company” and its ”Research and Technology Center” are most relevant to the main topic. Here we call ”Boeing Company” and ”Research and Technology Center” the focuses. Clearly, focused named entities are important for representing the main topic of the content. In the following, we show that the concept of focused named entity is useful for many natural language processing applications, such as summarization, search ranking and topic detection and tracking.

2.1 Using focused named entity for summarization

We consider the task of automatic summarization of the sample document in Figure 1. A traditional method is to select sentences with highest weights, where sentence weights are calculated by averaging term frequencies of words it contains. The resulting summarization is given in Figure 2. Using focused named entities, we consider two methods to refine the above summarization. The first method is to increase the weight of the focused named entity ”Boeing” in the sentences, leading to the summary in Figure 3. The other method simply picks sentences containing the focused named entity ”Boeing” as in Figure 4. From this example, we can see that summarization using focused named entities gives more indicative description of an article.

2.2 Using focused named entity for ranking search results

Suppose we want to find news about World Cup football match from a collection of news articles. First we search



☑ Boeing 5	☐ SEATTLE 1	☐ Boeing Commercial Airplanes 1	☑ Research and Technology Center 2
☐ Madrid 3	☐ Spain 1	☐ Miguel Hernan 1	☐ Dave Daggett 1
☐ Daggett 1	☐ Europe 2	☐ NASA 1	☐ Mike Friend 1

Boeing To Explore Electric Airplane

SEATTLE, Nov. 27, 2001 -- Boeing Commercial Airplanes will develop and test an electrically powered demonstrator airplane as part of a study to evaluate environmentally friendly fuel cell technology for future Boeing products.

Figure 1: Sample document with focused named entities marked

Boeing To Explore Electric Airplane

Fuel cells and electric motors will not replace jet engines on commercial transports, but they could one day replace gas turbine auxiliary power units.

Unlike a battery, which needs to be recharged, fuel cells keep working as long as the fuel lasts.

”Fuel cells show the promise of one day providing efficient, essentially pollution-free electrical power for commercial airplane primary electrical power needs,” Daggett said.

Figure 2: Summary using term frequency weighting

Boeing To Explore Electric Airplane

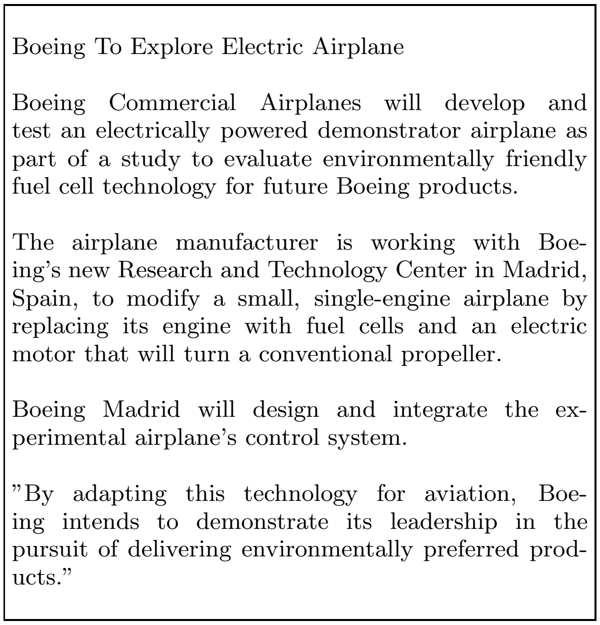
Boeing Commercial Airplanes will develop and test an electrically powered demonstrator airplane as part of a study to evaluate environmentally friendly fuel cell technology for future Boeing products.

Fuel cells and electric motors will not replace jet engines on commercial transports, but they could one day replace gas turbine auxiliary power units.

”By adapting this technology for aviation, Boeing intends to demonstrate its leadership in the pursuit of delivering environmentally preferred products.”

Figure 3: Summary weighted by focused named entities

[1] The original article can be accessed at http://www.boeing.com/news/releases/2001/q4/nr_011127a.html.



Boeing To Explore Electric Airplane

Boeing Commercial Airplanes will develop and test an electrically powered demonstrator airplane as part of a study to evaluate environmentally friendly fuel cell technology for future Boeing products.

The airplane manufacturer is working with Boeing's new Research and Technology Center in Madrid, Spain, to modify a small, single-engine airplane by replacing its engine with fuel cells and an electric motor that will turn a conventional propeller.

Boeing Madrid will design and integrate the experimental airplane's control system.

"By adapting this technology for aviation, Boeing intends to demonstrate its leadership in the pursuit of delivering environmentally preferred products."

Figure 4: Summary using sentences containing focused named entities

for documents containing the key phrase "World Cup". The ranking function, which determines which document is more relevant to the query, is very important to the search quality.

Since our query is a single phrase, the ranked search results, displayed in Table 1, are based on the term frequency of the phrase "World Cup". It is clear that without deeper text understanding, term frequency is a quite reasonable measure of relevancy. However, although some articles may contain more "World Cup" than others, they may actually focus less on the World Cup event which we are interested. Therefore a better indication of document relevancy is whether a document focuses on the entity we are interested in. A simple method is to re-order the search results first by whether the query entity is focused or not, and then by its term-frequency. It is quite clear that this method gives higher quality ranking.

In this example, we use Chinese corpus for demonstration, so the original searching results are in Chinese, which we have translated into English for reading convenience.

2.3 Other uses of focused named entity

We believe that focused named entities are also helpful in text clustering and categorization tasks such as topic detection and tracking. This is because if focused named entities are automatically recognized, then the event for each document can be described more precisely. Since focused named entities characterize what an article talks about, it is natural to organize articles based on them. Therefore by giving more weights to focused named entities, we believe that we can potentially obtain better quality clustering and more accurate topic detection and tracking.

Our study of the focused named entity recognition problem is motivated by its potential applications as illustrated above. Experiments in section 5.2 indicate that there is a sufficient agreement on focused named entities among human reviewers. Therefore our goal is to build a system that can automatically detect focused named entities among all named entities in a document. We shall mention that although this paper only studies named entities, the basic idea can be extended to tasks such as finding important words, noun-phrases in a document.

3. LEARNING BASED FOCUSED NAMED ENTITY RECOGNITION

Focused named entity recognition can be regarded as a binary classification problem. Consider the set of all named entities in a document extracted by a named entity recognition system. Each entity in this set can be labeled yes if it is a focused entity, or no if it is not. We formally define a two-class categorization problem as one to determine a label $y \in \{-1, 1\}$ associated with a vector x of input variables.

However, in order to build a successful focused named entity extractor, a number of issues have to be studied. First, named entities that refer to the same person or organization need to be grouped together; secondly what features are useful; and thirdly, how well different learning algorithms perform on this task. These issues will be carefully studied.

3.1 Coreference Resolution

Coreference is a common phenomenon in natural language. It means that an entity can be referred to in different ways and in different locations of the text. Therefore for focused named entity recognition, it is useful to apply a coreference resolution algorithm to merge entities with the same referents in a given document. There are different kinds of coreference according to the basic coreference types, such as pronominal coreference, proper name coreference, apposition, predicate nominal, etc. Here in our system, we only consider proper name coreference, which is to identify all variations of a named entity in the text.

Although it is possible to use machine learning methods for coreference resolution (see [20] as an example), we shall use a simpler scheme, which works reasonably well. Our coreference resolution method can be described as follows.

1. Partitioning: The set of named entities is divided into sub-sets according to named entity types, because coreference only occurs among entities with the same types.

2. Pair-wise comparison: Within each sub-set, pair-wise comparison is performed to detect whether each entity-pair is an instance of coreference. In this study, we use a simple algorithm which is based on string-matching only. Since we work with Chinese data, we split each entity into single Chinese characters. We study two different schemes here: using either exact string matching or partial string matching to decide coreference. In the case of exact string matching, two entities are considered to be a coreference pair only when they are identical. In the case of partial string matching, if characters in the shorter entity form a (non-consecutive) sub-string of the longer entity, then the two entities are considered to be a coreference pair.

3. Clustering: Merge all coreference pairs created in the second step into the same coreference chains. This step can also be done differently. For example, by using a sequential clustering method.

Although the coreference resolution algorithm described above is not perfect, it is not crucial since the results will

Table 1: Search result of ”World Cup”

focus/not	tf	title
focus	20	Uncover the Mystery of World Cup Draws
focus	11	Brazil and Germany Qualified, Iran Kicked out
focus	9	Preparing for World Cup, China Football Federation and Milutinovic Snatch the Time
focus	6	Sun Wen Understands the Pressure Milutinovic and China Team Faced
focus	5	Korea Leaves More Tickets to China Fans
focus	4	Paraguay Qualified, but Head Coach Dismissed
no	4	LiXiang: Special Relationships between Milutinovic and I
no	3	Three Stars on Golden Eagle Festival
focus	3	Adidas Fevernova, the Official 2002 FIFA World Cup Ball, Appears Before the Public in Beijing
no	2	China’s World Top 10 Start to Vote
focus	2	Qualified vs. Kicked out: McCarthy Stays on, Blazevic Demits
focus	2	China Attends Group Match in Korea, But not in the Same Group With Korea
no	2	Don’t Scare Peoples with Entering WTO
no	1	Kelon Tops China’s Home Appliance Industry in CCTV Ads Bidding
no	1	Lou Lan: Great Secrets Behind
focus	1	Australia Beats Uruguay by One Goal
no	1	Chang Hong’s ”King of Precision Display”: Good Friends of Football Fans

be passed to a machine learning algorithm in a later stage, which can offset the mistakes made in the coreference stage. Our experiment shows that by using coreference resolution, the overall system performance can be improved appreciably.

3.2 Classification methods

In this paper, we compare three methods: a decision tree based rule induction system, a Naive Bayes classifier, and a regularized linear classification method based on robust risk minimization.

3.2.1 Decision tree

In text-mining application, model interpretability is an important characteristic to be considered in addition to the accuracy achieved and the computational cost. The requirement of interpretability can be satisfied by using a rule-based system, such as rules obtained from a decision tree. Rule-based systems are particularly appealing since a person can examine the rules and modify them. It is also much easier to understand what a system does by examining its rules.

We shall thus include a decision tree based classifier in this study. In a typical decision tree training algorithm, there are usually two stages. The first stage is tree growing where a tree is built by greedily splitting each tree node based on a certain figure of merit. However after the first stage, the tree can overfit the training data, therefore a second stage involving tree pruning is invoked. In this stage, one removes overfitted branches of the tree so that the remaining portion has better predictive power. In our decision tree package, the splitting criteria during tree growth is similar to that of the standard C4.5 program [18], and the tree pruning is done using a Bayesian model combination approach. See [6] for detailed description.

3.2.2 Naive Bayes

Another very popular binary classification method is naive Bayes. In spite of its simplicity, it often achieves reasonable performance in practical applications. It can be regarded as a linear classification method, where we seek a weight vector w and a threshold θ such that $w^T x < \theta$ if its label $y = -1$ and $w^T x \geq \theta$ if its label $y = 1$. A score of value $w^T x - \theta$ can be assigned to each data point as a surrogate for the likelihood of x to be in class.

In this work, we adopt the multinomial model described in [14]. Let $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ be the set of training data. The linear weight w is given by $w = w^1 - w^{-1}$, and $\theta = \theta^1 - \theta^{-1}$. Denote by $x_{i,j}$ the j-th component of the data vector x_i, then the j-th component w_j^c of w^c $(c = \pm 1)$ is given by

$$w_j^c = \log \frac{\lambda + \sum_{i:y_i=c} x_{i,j}}{\lambda d + \sum_{j=1}^{d} \sum_{i:y_i=c} x_{i,j}},$$

and θ^c $(c = \pm 1)$ is given by $\theta^c = -\log \frac{|\{i:y_i=c\}|}{n}$.

The parameter $\lambda > 0$ in the above formulation is a smoothing (regularization) parameter. [14] fixed λ to be 1, which corresponds to the Laplacian smoothing.

3.2.3 Robust Risk Minimization Method

Similar to Naive Bayes, this method is also a linear prediction method. Given a linear model $p(x) = w^T x + b$, we consider the following prediction rule: predict $y = 1$ if $p(x) \geq 0$, and predict $y = -1$ otherwise. The classification error (we shall ignore the point $p(x) = 0$, which is assumed to occur rarely) is

$$I(p(x), y) = \begin{cases} 1 & \text{if } p(x)y \leq 0, \\ 0 & \text{if } p(x)y > 0. \end{cases}$$

A very natural way to compute a linear classifier is by finding a weight $(\hat{w}, \hat{b})$ that minimizes the average classification error in the training set:

$$(\hat{w}, \hat{b}) = \arg\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} I(w^T x_i + b, y_i).$$

Unfortunately this problem is typically NP-hard computationally. It is thus desirable to replace the classification error loss $I(p, y)$ with another formulation that is computationally more desirable. Large margin methods such as SVM employ modified loss functions that are convex. Many loss functions work well for related classification problems such as text-categorization [24, 10]. The specific loss function consider here is

$$h(p, y) = \begin{cases} -2py & py < -1 \\ \frac{1}{2}(py-1)^2 & py \in [-1, 1] \\ 0 & py > 1. \end{cases}$$

That is, our linear weights are computed by minimizing the following average loss on the training data:

$$(\hat{w}, \hat{b}) = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} h(w^T x_i + b, y_i).$$

This method, which we refer to as RRM (robust risk minimization), has been applied to linguistic processing [23] and text categorization [2] with good results. Detailed algorithm was introduced in [22].

4. FEATURES

We assume that named entities are extracted by a named entity recognition system. Many named entity recognition techniques have been reported in the literal, most of them use machine learning approach. An overview of these methods can be found in [19]. In our system, for the purpose of simplicity, we use human annotated named entities in the experiments. In the learning phase, each named entity is considered as an independent learning instance. Features must reflect properties of an individual named entity, such as its type and frequency, and various global statistical measures either at the document scale or at the corpus scale. This section describes features we have considered in our system, our motivations, and how their values are encoded.

4.1 Entity Type

Four entity types are defined: person, organization, place, and proper nouns. The type of a named entity is a very useful feature. For example, person and organization are more likely to be the focus than a place. Each entity type corresponds to a binary feature-component in the feature vector, taking a value of either one or zero. For example, a person type is encoded as [1 0 0 0], and an organization type is encoded as [0 1 0 0].

4.2 In Title or Not

Whether a named entity appears in the title or not is an important indicator of whether it is a focused entity. This is because title is a concise summary of what an article is about. The value of this feature is binary (0 or 1).

4.3 Entity Frequency

This feature is the number of times that the named entity occurs in the document. Generally speaking, the more frequent it occurs, the more important it is. The value of this feature is just the frequency of the named entity.

4.4 Entity Distribution

This feature is somewhat complicated. The motivation is that if a named entity occurs in many different parts of a document, then it is more likely to be an important entity. Therefore we use the entropy of the probability distribution that measures how evenly an entity is distributed in a document.

Consider a document which is divided into m sections. Suppose that each named entity's probability distribution is given by $\{p_1, p_2, ..., p_i, ..., p_m\}$, where $p_i = \frac{\text{occurrence in the } i\text{th section}}{\text{total occurrence in the document}}$. The entropy of the named entity distribution is computed by $entropy = -\sum_{i=1}^{m} p_i \log p_i$. In our experiments, we select $m = 10$. This feature contributes a real valued feature-component to the feature vector.

4.5 Entity Neighbor

The context in which a certain named entity appears is quite useful. In this study, we only consider a simple feature which counts its left and right neighboring entity types. If several named entities of the same type are listed side by side, then it is likely that the purpose is for enumeration, and the listed named entities are not important. Each neighboring side has five possible types — four named entity types plus a normal-word (not a named entity) type. For example, consider a person mentioned three times in the document. Among the three mentions, the left neighbors are two person names and one common word, and the right neighbors are one place name and two common words. Then the entity neighbor feature components are [2 0 0 0 1 0 0 1 0 2].

4.6 First Sentence Occurrence

This feature is inspired by the position method [3, 12] in sentence extraction. Its value is the occurrences of the entity appearing in the beginning sentence of a paragraph.

4.7 Document Has Entity in Title or Not

This feature indicates whether any entity exists in the title of the document, and thus takes binary value of 0 or 1.

4.8 Total Entity Count

This feature is the total number of entities in the document, which takes integer value. The feature reflects the relative importance of an entity in the entity set.

4.9 Document Frequency in the Corpus

This is a corpus level feature. If a named entity has a low frequency in the document collection, but relatively high frequency in the current document, then it is likely to be a focused entity. When this feature is used, the term frequency feature in section 4.3 will be computed using $(tf/docsize) * log(N/df)$, where df is the number of documents that a named entity occurs in.

5. EXPERIMENTS

In this section, we study the following issues: corpus annotation, human-level agreement on focused named entities, performance of machine learning methods compared with a baseline, influence of different features, and the impact of coreference module to the overall performance.

5.1 Corpus Annotation

We select fifteen days of Beijing Youth Daily news in November 2001 as our testing corpus, which contains 1,325 articles. The text, downloaded from http://bjyouth.ynet.com, is in Chinese. The articles belong to a variety of categories, including politics, economy, laws, education, science, entertainments, and sports.

Since different people may have different opinions on the focused named entities, a common set of rules should be agreed upon before the whole corpus is to be annotated. We use the following method to come up with a general guideline for annotating focused named entities.

First, the named entities in each document were annotated by human. Then, we selected twenty documents from the corpus and invited twelve people to mark focused named entities. Nine of the twelve people are experts in natural language processing, so their opinions are very valuable to define focused named entities. Based on the survey result, entities marked by more than half of the survey participants were defined as focused named entities. We obtained fifty focused named entities for the twenty articles. By studying the focused named entities in those articles, we were able to design specifications for focused named entity annotation. The whole corpus was then marked according to the specifications.

5.2 Human agreement statistics

In our survey, fifty entities were identified as focused entities from the total number of 341 entities in the 20 documents. Table 2 shows, among the 50 focused entities, 5 entities are agreed as focus by all 12 persons. and 7 entities are agreed by 11 persons, etc.

Table 2: Human Agreement Statistics

num of focused named entities	5	7	5	8	7	10	8
num of person agreeing	12	11	10	9	8	7	6

Let N_k denotes the number of person with agreement on focused named entity k, then the human agreement level $Agree_k$ on the k-th focused named entity is $Agree_k = \frac{N_k}{12}$. The average agreement on the 50 focused named entities is $Average\_Agree = \frac{\sum_{k=1}^{50} Agree_k}{50} = 72.17\%$, with variance 2.65%. We also computed the precision and the recall for the survey participants with respect to the fifty focused named entities. Table 3 shows that the best human annotator achieves an F_1 measure of 81.32%. Some of the participants marked either too many or too few named entities, and thus had much lower performance numbers. This problem was fixed when the whole corpus was annotated using specifications induced from this small-scale experiment.

Table 3: Human Annotation Performance

user id	precision	recall	F_1
1	90.24	74.00	81.32
2	86.05	74.00	79.57
3	83.33	70.00	76.09
4	84.21	64.00	72.73
5	96.55	56.00	70.89
6	90.63	58.00	70.73
7	71.74	66.00	68.75
8	73.81	62.00	67.39
9	57.14	80.00	66.67
10	48.19	80.00	60.15
11	38.60	88.00	53.66
12	33.33	94.00	49.21

5.3 Corpus Named Entity Statistics

We consider two data sets in our experiments: one is the whole corpus of 1,325 articles, and the other is a subset of 726 articles with named entities in their titles. Table 4 shows there are totally 3,001 focused entities among 18,371 entities in the whole corpus, which means that 16.34 percent of the entities are marked as focused. On average, there are 2.26 focused named entities for each article, which is consistent with the small-scale survey result.

Table 4: Corpus Statistics on Named Entities

set	docnum	entities	focuses	focus percent	focus/doc
1	1,325	18,371	3,001	16.34%	2.26
2	726	10,697	1,669	15.60%	2.30

5.4 Baseline results

Since named entities in title or with high frequency are more likely to be the focal entities, we consider three baseline methods. The first method marks entities in titles to be the foci; the second method marks most frequent entities in each article to be the focal entities; the third method is a combination of the above two, which selects those entities either in title or occurring most frequently. We use partial string matching for coreference resolution in the three baseline experiments.

Named entities occurring in the title are more likely to be the focus of the document, but they only represent a small portion of all focal entities. Baseline experiment 1 shows the precision of this method is quite high but the recall is very low.

Baseline experiment 2 implies that most of the top 1 named entities are focused entities, but again the recall is very low. However, if more named entities are selected, the precision is decreased significantly, so that the F_1 measure does not improve. The top-3 performance is the worst, with an F_1 measure of only 50.47%. Note that several named entities may have the same occurrence frequency in one document, which introduces uncertainty into the method.

By combining named entities from the title and with high frequency, we obtain better results than either of the two basic baseline methods. The best performance is achieved by combining the in-title and top 1 named entities, which achieves F_1 measures of 66.68% for data set 1, and 70.51% for data set 2.

5.5 Machine Learning Results

Since in our implementation, decision tree and naive Bayes methods only take integer features, we encode the floating features to integer values using a simple equal interval binning method. If a feature x is observed to have values bounded by x_{min} and x_{max}, then the bin width is computed by $\delta = \frac{x_{max} - x_{min}}{k}$ and the bin boundaries are at $x_{min} + i\delta$ where $i = 1, ..., k-1$. The method is applied to each continuous feature independently and k is set to 10. Although more sophisticated discretization methods exist, the equal interval binning method performs quite well in practice.

Machine learning results are obtained from five-fold cross-validation. Coreference resolution is done with partial string-matching. The test results are reported in Table 6.

This experiment shows that good performance can be achieved by using machine learning techniques. The RRM performance on both data sets are significantly better than the base line results, and comparable to that of the best human annotator we observed from our small-scale experiment in Section 5.2.

5.6 Influence of features

Table 5: Baseline Results

Corpus	Method	Focuses	focus/doc	Precision	Recall	F_1
726docs	title	992	1.36	83.47	49.61	62.23
1,325docs	top1	1,580	1.19	88.54	46.62	61.08
	top2	4,194	3.17	54.48	76.14	63.52
	top3	7,658	5.78	35.13	89.64	50.47
726docs	title+top1	1,247	1.72	82.44	61.59	70.51
	title+top2	2,338	3.22	56.93	79.75	66.43
	title+top3	4,165	5.74	36.06	89.99	51.49
1,325docs	title+top1	2,011	1.52	83.09	55.68	66.68
	title+top2	4,388	3.31	53.78	78.64	63.88
	title+top3	7,738	5.84	34.94	90.10	50.36

Table 6: Machine Learning Results

Dataset	RRM			Decision Tree			Naive Bayes		
	P	R	F_1	P	R	F_1	P	R	F_1
726 docs	88.51	80.54	84.27	87.29	78.02	82.37	69.32	90.28	78.37
1,325 docs	84.70	78.23	81.32	83.83	74.61	78.89	69.14	89.08	77.82

The goal of this section is to study the impact of different features with different algorithms. Results are reported in Table 7. Feature id corresponds to the feature subsection number in section 4.

Experiment A uses frequency-based features only. It is quite similar to the bag-of-word document model for text categorization, with the entity-frequency and in-title information. By adding more sophisticated document-level features, the performance can be significantly improved. For the RRM method, F_1 finally reaches 81.32%. It is interesting to observe that the corpus-level feature (experiment F versus G) has different impacts on the three algorithms. It is a good feature for naive Bayes, but not for the RRM and decision tree. Whether corpus-level features can appreciably enhance the classification performance requires more careful investigation.

The experiments also indicate that the three learning algorithms do not perform equally well. RRM appears to have the best overall performance. The naive Bayes method requires all features to be independent, which is a quite unrealistic assumption in practice. The main problem for decision tree is that it easily fragments the data, so that the probability estimate at the leaf-nodes become unreliable. This is also the reason why voted decision trees (using procedures like boosting or bagging) perform better.

The decision tree can find rules readable by a human. For example, one such rule reads as: if a named entities appears at least twice, its left and right neighbors are normal words, its discrete distribution entropy is greater than 2, and the entity appears in the title, then the probability of it being a focused entity is 0.87.

5.7 Coreference Resolution

In order to understand the impact of coreference resolution on the performance of focused named entity recognition, we did the same set of experiments as in section 5.5, but with exact string matching only for coreference resolution in the feature extraction process. Table 8 reports the five-fold cross validation results. On average the performance is decreased by about 3 to 5 percent. This means coreference resolution plays an important role in the task. The reason is that it maps variations of a named entity into a single group, so that features such as occurrence frequency and entity distribution can be estimated more reliably. We believe that with more sophisticated analysis such as pronominal coreference resolution, the classification performance can be further improved.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we studied the problem of focused named entity recognition. We gave examples to illustrate that focused named entities are useful for many natural language processing applications. The task can be converted into a binary classification problem. We focused on designing linguistic features, and compared the performance of three machine learning algorithms. Our results show that the machine learning approach can achieve near human-level accuracy. Because our system is trainable and features we use are language independent, it is easy for us to build a similar classification model for other languages. Our method can also be generalized to related tasks such as finding important words and noun-phrases in a document.

In the future, we will integrate focused named entity recognition into real applications, such as information retrieval, automatic summarization, and topic detection and tracking, so that we can further study and evaluate its influences to these systems.

Acknowledgments

We thank Honglei Guo for providing the original corpus with named entity annotation. Jianmin Jiang helped us to set up named entity annotation server, which made it much easier to view and annotate the corpus. We thank ZhaoMing Qiu, ShiXia Liu, ZhiLi Guo for their valuable comments on the experiments. We are also grateful to our colleagues who spent their time to participate in the focused named entity survey.

7. REFERENCES

[1] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proceedings of the ACL*

Table 7: Performance of different features with Different Algorithms

ID	Features	RRM			Decision Tree			Naive Bayes		
		P	R	F_1	P	R	F_1	P	R	F_1
A	2+3+7	79.11	20.86	32.96	77.48	61.81	68.70	96.39	33.47	49.67
B	1+2+3	71.95	82.08	76.60	71.06	72.31	71.23	93.29	42.91	58.76
C	1+2+3+7	73.32	0.8143	76.87	70.90	78.63	74.54	92.58	48.65	63.74
D	1+2+3+7+5	70.60	84.99	76.98	74.42	75.85	75.09	85.44	61.96	71.71
E	1+2+3+7+5+8	86.15	75.89	80.68	74.42	75.85	75.09	66.56	86.14	75.07
F	$1+2+\cdots+7+8$	85.98	77.37	81.44	79.62	78.30	78.92	66.40	89.44	76.19
G	$1+2+\cdots+8+9$	84.70	78.23	81.32	83.83	74.61	78.89	69.14	89.08	77.82

Table 8: Machine Learning Test Result with exact string-matching for Coreference Resolution

data set	RRM			Decision Tree			Naive Bayes		
	P	R	F_1	P	R	F_1	P	R	F_1
726 docs	84.43	75.21	79.49	83.13	73.68	78.10	67.85	85.64	75.64
1,325 docs	81.67	72.60	76.74	79.60	70.45	74.69	66.77	83.56	74.20

Intelligent Scalable Text Summarization Workshop (ISTS'97), pages 10–17, 1997.

[2] F. J. Damerau, T. Zhang, S. M. Weiss, and N. Indurkhya. Text categorization for a comprehensive time-dependent benchmark. *Information Processing & Management*, 2004.

[3] H. P. Edmundson. New methods in automatic abstracting. *Journal of The Association for Computing Machinery*, 16(2):264–285, 1969.

[4] J. Y. Ge, X. J. Huang, and L. Wu. Approaches to event-focused summarization based on named entities and query words. In *DUC 2003 Workshop on Text Summarization*, 2003.

[5] E. Hovy and C.-Y. Lin. Automated text summarization in summarist. In I. Mani and M. Maybury, editors, *Advances in Automated Text Summarization*, pages 81–94. MIT Press, 1999.

[6] D. E. Johnson, F. J. Oles, T. Zhang, and T. Goetz. A decision-tree-based symbolic rule induction system for text categorization. *IBM Systems Journal*, 41:428–437, 2002.

[7] M.-Y. Kan and K. R. McKeown. Information extraction and summarization: domain independence through focus types. Columbia University Computer Science Technical Report CUCS-030-99.

[8] J. M. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *SIGIR '95*, pages 68–73, 1995.

[9] D. Lawrie, W. B. Croft, and A. Rosenberg. Finding topic words for hierarchical summarization. In *SIGIR '01*, pages 349–357, 2001.

[10] F. Li and Y. Yang. A loss function analysis for classification methods in text categorization. In *ICML 03*, pages 472–479, 2003.

[11] C.-Y. Lin. Training a selection function for extraction. In *CIKM '99*, pages 1–8, 1999.

[12] C.-Y. Lin and E. Hovy. Identifying topics by position. In *Proceedings of the Applied Natural Language Processing Conference (ANLP-97)*, pages 283–290, 1997.

[13] D. Marcu. From discourse structures to text summaries. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 82–88. ACL, 1997.

[14] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI/ICML-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998.

[15] J. L. Neto, A. Santos, C. Kaestner, A. Freitas, and J. Nievola. A trainable algorithm for summarizing news stories. In *Proceedings of PKDD'2000 Workshop on Machine Learning and Textual Information Access*, September 2000.

[16] C. Nobata, S. Sekine, H. Isahara, and R. Grishman. Summarization system integrated with named entity tagging and ie pattern discovery. In *Proceedings of Third International Conference on Language Resources and Evaluation (LREC 2002)*, 2002.

[17] C. D. Paice and P. A. Jones. The identification of important concepts in highly structured technical papers. In *SIGIR '93*, pages 69–78. ACM, 1993.

[18] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[19] E. F. T. K. Sang and F. D. Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147, 2003.

[20] W.-M. Soon, H.-T. Ng, and C.-Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

[21] S. Teufel and M. Moens. Sentence extraction as a classification task. In *ACL/EACL-97 Workshop on Intelligent and Scalable Text Summarization*, 1997.

[22] T. Zhang. On the dual formulation of regularized linear systems. *Machine Learning*, 46:91–129, 2002.

[23] T. Zhang, F. Damerau, and D. E. Johnson. Text chunking based on a generalization of Winnow. *Journal of Machine Learning Research*, 2:615–637, 2002.

[24] T. Zhang and F. J. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval*, 4:5–31, 2001.