Ranking Definitions with Supervised Learning Methods

Jun Xu*
College of Software
Nankai University,
No.94 Weijin Road,
Tianjin, China, 300071
nkxj@yahoo.com.cn

Yunbo Cao
Microsoft Research Asia
5F Sigma Center,
No.49 Zhichun Road,
Haidian Beijing, China,
100080
yucao@microsoft.com

Hang Li
Microsoft Research Asia
5F Sigma Center,
No.49 Zhichun Road,
Haidian Beijing, China,
100080
hangli@microsoft.com

Min Zhao*
Institute of Automation
Chinese Academy of
Sciences, Beijing, China
m.zhao@mail.ia.ac.cn

ABSTRACT

This paper is concerned with the problem of definition search. Specifically, given a term, we are to retrieve definitional excerpts of the term and rank the extracted excerpts according to their likelihood of being good definitions. This is in contrast to the traditional approaches of either generating a single combined definition or simply outputting all retrieved definitions. Definition ranking is essential for the task. Methods for performing definition ranking are proposed in this paper, which formalize the problem as either classification or ordinal regression. A specification for judging the goodness of a definition is given. We employ SVM as the classification model and Ranking SVM as the ordinal regression model respectively, such that they rank definition candidates according to their likelihood of being good definitions. Features for constructing the SVM and Ranking SVM models are defined. An enterprise search system based on this method has been developed and has been put into practical use. Experimental results indicate that the use of SVM and Ranking SVM can significantly outperform the baseline methods of using heuristic rules or employing the conventional information retrieval method of Okapi. This is true both when the answers are paragraphs and when they are sentences. Experimental results also show that SVM or Ranking SVM models trained in one domain can be adapted to another domain, indicating that generic models for definition ranking can be constructed.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process*; H.4.m [**Information Systems and Applications**]: Miscellaneous; I.7.m [**Document and Text Processing**]: Miscellaneous; H.5 [**Information Systems Applications**]: Information Interfaces and Presentation

General Terms

Algorithms, Experimentation, Human Factors.

Keywords

Web search, search of definitions, text mining, web mining, ordinal regression, and classification.

1. INTRODUCTION

People will find it helpful, if we develop a system that can automatically find definitions of terms from documents on the web (either Internet or intranet). This is because definitions describe the meanings of terms and thus belong to the type of frequently accessed information.

Traditional information retrieval is designed to search for relevant documents (e.g., [15]), and thus is not suitable for performing the task.

TREC formalizes the problem as that of definitional question answering [19, 20]. Given the questions of "what is X" or "who is X", one extracts answers from multiple documents and combines the extracted answers into a single unified answer (e.g., [4, 6, 7, 21, 23]). Question answering is ideal as a means of helping people find definitions. However, it might be difficult to realize it in practice. Usually definitions extracted from different documents describe the term from different perspectives (as will be discussed in Section 3), and thus it is not easy to combine them together.

Methods for extracting definitions from documents have also been proposed in text mining. All of the methods resort to human-defined rules for definition extraction and do not consider ranking of definitions [10, 13].

In this paper, we consider a problem of what we call 'definition search'. More specifically, given a query term, we automatically extract all likely definition candidates about the term (paragraphs or sentences) from documents and rank the definition candidates according to the degrees of being good definitions.

Definition ranking is essential for the task. We formalize the problem of definition ranking as either that of classification between good and bad definitions, or that of ordinal regression among good, bad and indifferent definitions. We propose a specification for judging whether a definition is a 'good', 'bad', or 'indifferent' definition. We employ SVM and Ranking SVM models as our classification and ordinal regression models respectively. We also develop features used in the SVM and Ranking SVM models. We perform definition ranking in the following way. First, we use heuristic rules to select likely definition candidates; second, we employ SVM or Ranking SVM models to rank the candidates; and third, we remove those redundant candidates staring from the top of the ranked list. We then store the ranked definitions for each term. In search, we return the ranked definitions on a given term.

Our experimental results indicate that our approach is significant for definition ranking. We show that good definitions are often ranked higher using our approach than using baseline methods. We have also constructed a large-scale search system on the basis of the proposed approach and have empirically verified its effectiveness. Other experimental findings are that the trained

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

* The work was conducted when Xu and Zhao were visiting Microsoft Research Asia.

models can be generic in the sense that they are almost domain independent and that the approach can be applied to both sentence level and paragraph level.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 advocates the necessity of conducting research on definition ranking. Section 4 gives a specification on goodness of definitions. Section 5 explains our approach to definition ranking. Section 6 introduces the use of definition ranking in search of definitions, and Section 7 describes a system providing the definition search function. Section 8 reports our experimental results. Section 9 summarizes our work in the paper.

2. RELATED WORK

2.1 Automatically Discovering Definitions

Google offers a feature of definition search [24]. When a user types "define:<*term*>" in the search box, the search engine returns glossaries containing the definitions of <*term*>. This feature relies on the fact that there are many glossary web pages available on the Internet. While it is not clear how Google collects the glossary web pages, it seems that the pages have common properties. The titles of the pages usually contain the words 'glossary', 'dictionary' etc; the terms in a page are sorted in alphabetic order; and the definitions in a page are usually presented in the same format (e.g., terms are highlighted in boldface).

TREC has a task of definitional question answering. In the task, "what is <*term*>" and "who is <*person*>" questions are answered in a single combined text [19, 20]. Results of question answering are evaluated by humans.

Systems have been developed for performing the question answering task of TREC. In TREC 2003, most of the systems [4, 6, 7, 21, 23] employed both statistical learning methods and human defined rules. They assumed that in addition to the corpus data in which the answers can be found, there are other data available such as web data (with Google as search engine) and encyclopedia data. They attempted to use the extra data to enhance the quality of question answering.

For instance, the system developed by BBN [21] performs definitional question answering in six steps. First, the system identifies which type the question is: who type or what type. Second, it collects all documents relevant to the question term from the TREC corpus using information retrieval technologies. Third, it pinpoints the sentences containing the question term in the retrieved documents using heuristic rules. Fourth, it harvests the kernel facts about the question term using language processing and information extraction technologies. Fifth, it ranks all the kernel facts by their importance and their similarities to the profile of the question term. Finally, it generates an answer from the non-redundant kernel facts with heuristic rules.

Text mining methods have also been proposed which can employ human-defined rules (patterns) to extract terms and their definitions.

For instance, DEFINDER [10] is a system that mines definitions from medical documents. The system consists of two modules. One module utilizes a shallow finite state grammar to extract definitions. The other module makes use of a deep dependency grammar to extract definitions. The system combines the extracted results of the two modules.

Liu et al propose a method of mining topic-specific knowledge on the web [13]. They extract information such as definitions and sub-topics of a specific topic (e.g., data mining) from the web. In definition extraction, they make use of manually defined rules containing linguistic information as well as HTML information.

For other work on definition discovery, see also [1, 2, 3, 5, 16].

2.2 Ordinal Regression

Ordinal regression (or ordinal classification) is a problem in which one classifies instances into a number of ordered categories. It differs from classification in that there is a total order relationship between the categories. Herbrich et al [8] propose an algorithm for conducting this task.

Joachims [9] proposes learning a ranking function for search as ordinal regression using click-through data. He employs what he calls the Ranking SVM model for ordinal regression.

Tan et al [17] show another example of viewing search as ordinal regression.

3. Definition Search

First, let us describe the problem of 'definition search' more precisely. As input, we first receive a query term. The query term is usually a noun phrase representing a concept. We automatically extract all likely definition candidates from the document collection. The candidates can be either paragraphs or sentences. Next, we rank the definition candidates according to the degree to which each one is a good definition and output them.

Without loss of generality, in this paper we only consider definitions of technical terms, i.e., we do not consider definitions of persons.

Next, let us explain why the problem setting has value in practice.

Definition search can be useful in different information retrieval scenarios, for example, definition search at a company intranet. We have conducted a survey at an IT company in which we ask the employees what kind of searches they have ever performed on their company intranet. Figure 1 shows the result of one question. We see that 77% of the people have experiences of searching for "what is" questions.

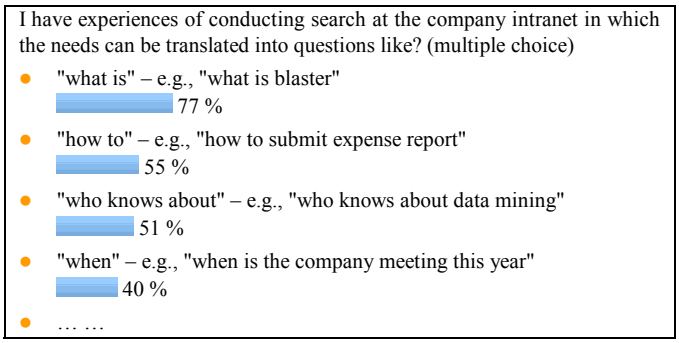

Figure 1: A survey on experiences of search in an IT Company.

Google's approach to finding definitions has an advantage: the quality of the retrieved definitions is high. However, it also has a limitation: it is based on the assumption that there are many high quality glossaries available. This is true for the Internet, but is not necessarily true for an extranet or an intranet.

1. HTML is an application of ISO Standard 8879:1986 Information Processing Text and Office Systems; Standard Generalized Markup Language (SGML). The HTML Document Type Definition (DTD) is a formal definition of the HTML syntax in terms of SGML.
2. HTML is an acronym for Hyper Text Markup Language, which is the standard that defines how Web documents are formatted. HTML is a subset of SGML, which is the acronym for Standardized General Markup Language.
3. HTML is a text-based programming language that uses tags to tell the browser how to display information such as text and graphics.
4. HTML is the programming language used to write Web pages. It defines a formatting syntax, based on tags, for displaying pages of information, for example font, font size, back ground color, image placement and so on.

Figure 2: Definitions of 'HTML' from different perspectives

1. Linux is an open source operating system that was derived from UNIX in 1991.
2. Linux is a UNIX-based operating system that was developed in 1991 by Linus Torvalds , then a student in Finland.
3. Linux is a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world.
4. Linux is a command line based OS.
5. Linux is the best-known product distributed under the GPL.
6. Linux is the platform for the communication applications for the dealer network.
7. Linux is a Unicode platform.
8. Linux is an excellent product.
9. Linux is a threat to Microsoft's core businesses.

Figure 3: Example definition candidates for 'Linux'

We tried to collect glossaries at an intranet and found that there were only a few glossaries available. We collected all the web pages containing at least one of the key words 'glossary', 'gloss', 'dictionary', 'definition', or 'define' and manually checked whether they are glossary pages. From about 1,000,000 web pages in total, we were only able to find about 50 glossary pages containing about 1000 definitions.

We note that even for Google's approach, ranking of definitions is still necessary. For the query term of 'XML', for example, Google returns 25 definitions. It may not be necessary for people to look at all the definitions.

TREC's approach to finding definitions is ideal because it provides a single combined summary of the meaning of each term. One can get all the necessary information by reading the summary, if the summary is good enough. However, it is also challenging, as generation of such a summary is not easy, even not possible.

A term can be defined from different perspectives and the contents of the definitions extracted from different documents can be diverse. It is a difficult task (even for humans) to summarize them into a natural text. This is particularly true when the extracted definition candidates are paragraphs (cf., the example paragraphs in Figure 2).

We note that this also relates to the famous philosophical problem raised by Wittgenstein. He argues that usually there is no set of properties commonly shared by all the instances of a concept (e.g., 'game'), which can be used in definition of the concept [11].

Furthermore, the qualities of definitions extracted from different documents can vary. Usually, there are many definitions which can not be viewed as 'good definitions'. (A specification on good definition will be given in Section 4). However, they can still help people's understanding as 'explanations' of terms: they are especially useful when there are not enough good definitions found. Ranking can be used as a mechanism for users to look at likely definitions.

Figure 3 shows example sentences (excerpts) about the term 'Linux', which are extracted from real texts. Sentences 1-3 describe the general notion and the main properties of 'Linux', and thus can be viewed as good definitions. Sentences 4-7 explain the properties of 'Linux' each from one viewpoint and sentences 8-9 are opinions on 'Linux'. However, they still provide useful information.

Note that our approach is not contradictory to TREC's approach. Instead, ranking of definitions can be used as one step within the methods developed in TREC.

We should also note that there is another difference between our problem setting and the settings used in the TREC systems. That is, we do not assume here that additional data like encyclopedia data is available. This is because such data is not always available, particularly when it is on an intranet. (In our experiments described in Section 8, we used data from an intranet).

In the text mining methods described in Section 2.1, extracted definitions are treated uniformly and thus are not ranked. As we have discussed, however, definitions should be sorted in their likelihood of being good definitions. It makes sense, therefore, if we rank the extracted definitions and use only the top *n* good definitions. We can thus employ definition ranking as one step in the existing text mining methods.

4. SPECIFICATION OF GOODNESS OF DEFINITIONS

Judging whether a definition is good or not in an objective way is hard. However, we can still provide relatively objective guidelines for the judgment. We call it the specification in this paper. It is indispensable for development and evaluation of definition ranking.

In the specification, we create three categories for definitions which represent their goodness as definitions: 'good definition', 'indifferent definition' and 'bad definition'.

A good definition must contain the general notion of the term (i.e., we can describe the term with the expression "is a kind of") and several important properties of the term. From a good definition, one can understand the basic meaning of the term. Sentences 1-3 in Figure 3 are examples of a good definition.

A bad definition neither describes the general notion nor the properties of the term. It can be an opinion, impression, or feeling of people about the term. One cannot get the meaning of the term by reading a bad definition. Sentences 8-9 in Figure 3 are examples of a bad definition.

An indifferent definition is one that between good and bad definitions. Sentences 4-7 in Figure 3 are examples.

5. Definition Ranking

In definition ranking, we extract from the entire collection of documents <*term, definition, score*> triples. They are respectively term, a definition of the term, and its score representing its likelihood of being a good definition.

First, we collect definition candidates (paragraphs) using heuristic rules. That means that we filter out all unlikely candidates. Second, we calculate the score of each candidate as definition using a SVM or Ranking SVM. As a result, we obtain triples of <*term, definition, score*>. Third, we find similar definitions using Edit Distance and remove the redundant definitions. The SVM and Ranking SVM are trained in advance with labeled instances.

The first step can be omitted in principle. With the adoption of it, we can enhance the efficiency of both training and ranking

Both paragraphs and sentences can be considered as definition excerpts in our approach. Hereafter, we will only describe the case of using paragraphs. It is easy to extend it to the case of using sentences.

5.1 Collecting Definition Candidates

We collect from the document collection all the paragraphs that are matched with heuristic rules and output them as definition candidates.

First, we parse all the sentences in the paragraph with a Base NP (base noun phrase) parser and identify <*term*> using the following rules. (For the definition of Base NP, see for example [22].)

1. <*term*> is the first Base NP of the first sentence.
2. Two Base NPs separated by 'of' or 'for' are considered as <*term*>. For example, 'Perl for ISAPI' is the term from the sentence "Perl for ISAPI is a plug-in designed to run Perl scripts…"

In this way, we can identify not only single word <*term*>s, but also more complex multi-word <*term*>s.

Next, we extract definition candidates with the following patterns,

1. <*term*> is a|an|the *
2. <*term*>, *, a|an|the *
3. <*term*> is one of *

Here, '*' denotes a word string containing one or more words and '|'denotes 'or'.

Note that the step of collecting definition candidates is similar to the method of definition extraction employed in [13]. The uses of other sets of rules for candidate selection are also possible. However, they are not essential for conducting definition ranking. As mentioned above, we can skip this step or reinforce it by using more sophisticated rules.

5.2 Ranking Definition Candidates

Ranking definition candidates determines the goodness of a candidate as a definition. The goodness of a definition candidate is determined by the characteristic of the paragraph and is independent from the term itself. Thus, ranking on the basis of goodness as definition differs from ranking on the basis of relevance to query in traditional information retrieval.

We take a statistical machine learning approach to address the ranking problem. We label candidates in advance, and use them for training.

Let us describe the problem more formally. Given a training data set $D = \{x_i, y_i\}_1^n$, we construct a model that can minimize error in prediction of y given x (generalization error). Here $x_i \in X$ and $y_i \in \{good, indifferent, bad\}$ represent a definition candidate and a label, respectively. When applied to a new instance x, the model predicts the corresponding y and outputs the score of the prediction.

For ordinal regression, we employ Ranking SVM, and for classification we employ SVM. SVM or Ranking SVM assigns a score to *each* definition candidate. The higher the score, the better the candidate is as a definition.

5.2.1 Ranking based on Ordinal Regression

Classifying instances into the categories: 'good', 'indifferent' and 'bad' is a typical ordinal regression problem, because there is an order *between* the three categories. The cost of misclassifying a good instance into 'bad' should be larger than that of misclassifying the instance into 'indifferent'.

We employ Ranking SVM [9] as the model of ordinal regression. Given an instance x (definition candidate), Ranking SVM assigns a score to it based on

$$U(x) = w^T x, \quad (1)$$

where w represents a vector of weights. The higher the value of $U(x)$ is, the better the instance x is as a definition. In ordinal regression, the values of $U(x)$ are mapped into intervals on the real line and the intervals correspond to the ordered categories. An instance that falls into one interval is classified into the corresponding ordered category.

In our method of definition ranking, we only use scores output by a Ranking SVM.

The construction of a Ranking SVM needs labeled training data (in our case, the ordered categories are good, indifferent, and bad definitions). Details of the learning algorithm can be found in [9]. In a few words, the learning algorithm creates the so-called utility function in (1), such that the utility function best reflects the 'preference orders' between the instance pairs in the training data.

5.2.2 Ranking based on Classification

In this method, we ignore indifferent definitions and only use good and bad definitions. This is because indifferent definitions may not be important for the training of ranking on the basis of goodness, especially when a classification mode is used (our experimental results have also verified this). Therefore, we can address the problem as that of binary classification.

We employ SVM (Support Vector Machines) [18] as the model of classification. Given an instance x (definition candidate), SVM assigns a score to it based on

$$f(x) = w^T x + b, \quad (2)$$

where w denotes a vector of weights and b denotes a intercept. The higher the value of $f(x)$ is, the better the instance x is as a definition. In classification, the sign of $f(x)$ is used. If it is positive, then x is classified into the positive category, otherwise into the negative category.

In our method of definition ranking, we only use scores output by SVM for ranking.

The construction of SVM needs labeled training data (in our case, the categories are good and bad definitions). Details of the learning algorithm can be found in [18]. In a few words, the learning algorithm creates the 'hyper plane' in (2), such that the hyper plane separates the positive and negative instances in the training data with the largest 'margin'.

Both Ranking SVM and SVM can be extended to non-linear models based on kernel functions. In this paper, we only consider the uses of linear models.

5.2.3 Features

Ranking SVM and SVM utilize the same set of features. Table 1 shows the list of the features. There are positive features like (1) and (7). That is, if the term appears at the beginning of the paragraph or repeatedly occurs in the paragraph, then it is likely the paragraph is a definition on the term. There are also negative features like (4). If words like 'she', 'he', or 'said' occurs in the paragraph, it is likely the paragraph is not a (good) definition.

Ranking SVM and SVM also rely on 'bag-of-words' features. High frequency words appearing immediately after terms in training data are collected as keywords. If a paragraph contains such a keyword, then the corresponding feature value will be 1, otherwise 0.

Table 1: Features used in ranking models

1.	<*term*> occurs at the beginning of the paragraph.
2.	<*term*> begins with 'the', 'a', or 'an'.
3.	All the words in <*term*> begin with uppercase letters.
4.	Paragraph contains predefined negative words, e.g. 'he', 'she', 'said'
5.	<*term*> contains pronouns.
6.	<*term*> contains 'of', 'for', 'and', 'or' or ','.
7.	<*term*> re-occurs in the paragraph.
8.	<*term*> is followed by 'is a ', 'is an' or 'is the '.
9.	Number of sentences in the paragraph.
10.	Number of words in the paragraph.
11.	Number of the adjectives in the paragraph.
12.	Bag of words: words frequently occurring within a window after <*term*>

5.3 Removing Redundant Candidates

After ranking, we obtain a ranked list of definition candidates for each term. Usually there are duplicate (or partially duplicate) definition candidates. We should remove them because they are redundant for users.

We conduct duplicate candidate removal from the top of the ranked candidates. We determine whether two definition candidates are duplicates or partial duplicates using Edit Distance [12]. If two definition candidates are too similar, we remove the one whose score is lower.

6. Search of Definitions

In search of definitions, given a query term, we retrieve all the triples matched against the query term and present the corresponding definitions in descending order of the scores.

All the data necessary for definition search is stored in a database table in advance. The data is in the form of <*term, definition, score*> triples. For each term, the corresponding definition candidates and scores are grouped together and the definition candidates are sorted in descending order of the scores.

During search, we retrieve the sorted definition candidates with regard to the search term by table lookup. For example, given the query term 'Linux', we retrieve the ranked list of the definition candidates as those in Table 2.

Table 2: Ranked list of definitions for 'Linux'

Definition	Score
1. Linux is an open source operating system that was derived from UNIX in 1991.	1.9469
2. Linux is a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world.	1.6816
3. Linux is a UNIX-based operating system that was developed in 1991 by Linus Torvalds, then a student in Finland.	1.6289
4. Linux is the best-known product distributed under the GPL.	1.0206
5. Linux is the platform for the communication applications for the dealer network.	0.8764
6. Linux is a command line based OS.	0.7485
7. Linux is a Unicode platform.	0.6553
8. Linux is a phenomenon that is growing from the bottoms up.	0.3219
9. Linux is an excellent product.	0.1710

7. IMPLEMENTATION IN SEARCH SYSTEM

We have developed an enterprise search system and have put it into practical use at the intranet of an IT company. The system calledInformation Desk (cf., Figure 4) provides four types of search. Search of definitions is among them. The four features include:

1. 'what is' – search of definitions and acronyms. Given a term, it returns a list of definitions of the term. Given an acronym, it returns a list of possible expansions of the acronym.
2. 'who is' – search of employees' information. Given the name of a person, it returns his/her profile information, authored documents and associated key terms.
3. 'where are homepages of' – search of homepages. Given the name of a group, a product, or a technology, it returns a list of its related home pages.
4. 'who knows about' – search of experts. Given a term on a technology or a product, it returns a list of persons who might be experts on the technology or the product.

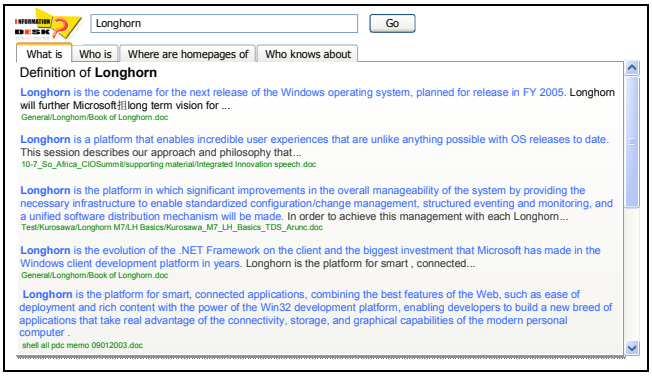

Figure 4: Information Desk system

In this paper, we only explain how the 'search of definitions' feature works. The details of other parts of system will be reported else where.

In the system, there are more than 2,000,000 documents crawled (including 980,000 HTML pages and 110,000 Word documents). We extracted from the documents about 50,000 definition candidates on about 31,000 terms. Rankings of the definitions have also been created using the method proposed in this paper. The terms are on products, services, projects, organizations, and technologies.

When a user searches for the definitions of a term, the system returns a ranked list of the definitions (candidates) of the term and the links of the documents containing the definitions. The user can not only get the definitions of the terms, but also get the original contexts of the definitions.

We have also asked the participants to the survey described in Section 3, which feature has helped them in finding information. 23% of the participants have replied that the feature of definition search is helpful (cf., Figure 5).

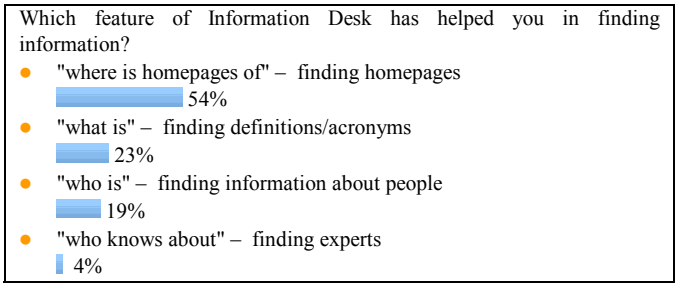

Figure 5: A survey result on Information Desk.

8. EXPERIMENTAL RESULTS

We have conducted experiments to verify the effectiveness of our proposed approach to definition ranking. Particularly, we have investigated whether ranking of definitions can be solved as ordinal regression or classification. We have conducted the experiments at two levels of granularity, namely, ranking paragraphs and sentences as definitions. We have also investigated whether the trained models are domain independent.

We did not try to use different sets of rules for collecting definition candidates, because they are not essential for evaluation of definition ranking methods.

8.1 Baseline and Measure for Evaluation

As one baseline method, we used Okapi [14]. Given a query term, it returns a list of paragraphs or sentences ranked only on the basis of relevance to the query term.

As another baseline method, we used random ranking of definition candidates. This can be viewed as an approximation of existing methods on definition extraction.

We made use of three measures for evaluation of definition ranking. They are 'error rate of preference pairs' (cf., [8, 9]), R-precision (precision of R highest ranked candidates, where R is number of 'good' definitions), and Top N precision (percentage of terms whose top N ranked candidates contain 'good' definitions. N = 1 or 3). Equations (3), (4) and (5) give the details.

$$Error\ rate = \frac{|msitakenly\ predicted\ preference\ pairs|}{|all\ preference\ pairs|} \quad (3)$$

$$R\text{-}precision(term_i) = \frac{|good\ definitions\ at\ \ R\ highest\ ranked\ candidates|}{R},$$

where R is number of good definitions for $term_i$.

$$R\text{-}precision = \frac{\sum_{i=1}^{T} R\text{-}precision(term_i)}{T}, \quad (4)$$

where T is number of terms in data set

$$Top\ N\ Precision = \frac{|terms\ whose\ top\ N\ ranked\ candidates\ contain\ 'good'|}{|all\ terms\ in\ data\ set|} \quad (5)$$

8.2 Ranking Definitional Paragraphs in Intranet Data

We crawled from the intranet of an IT company (referred to as 'Intranet' hereafter and constructed a document set.

In this experiment, we considered a paragraph as an instance for definition search. First, we extracted all the <*term, definition*> pairs from the crawled web documents as described in Section 7. Then we randomly selected about 200 distinct terms and their definition candidates. There were a number of terms having only one associated candidate: we removed these terms and candidates. After that, human annotators were asked to label the remaining candidates (as good, indifferent and bad definitions) following the specification described in Section 4. Finally, terms without good definition candidates were discarded.

Our final data set contains 95 terms and 1366 candidates. On average, each term has 2.37 (225/95) good definitions. Table 3 shows statistics on the data. We tested the effectiveness of ranking with both Ranking SVM and SVM using this data set.

We conducted 5-fold cross validation. The results reported in Table 4 are those averaged over the 5 trials.

In the experiment, for SVM, we used only the good and bad definitions in training data for training and used all of the definitions in test data for test. For Ranking SVM, we used all the definitions (good, indifferent and bad) in training and test data for training and test respectively. (We also tried using all the definitions in training data for training SVM. However, the results were not as good as the results of using only good and bad definitions).

Table 3: Statistics on Intranet paragraph data

Number of terms	95
Number of definition candidates	1366
Number of good definitions	225
Number of indifferent definitions	470
Number of bad definitions	671

Table 4: Definitional paragraph ranking on Intranet data

	Error Rate	R-Precision	Top 1 Precision	Top 3 Precision
Okapi	0.5133	0.2886	0.2211	0.6421
SVM	0.3284	0.4658	0.4324	0.8351
Ranking SVM	**0.2712**	**0.5180**	**0.5502**	**0.8868**
Random Ranking	0.4363	0.3224	0.3474	0.6316

Table 5: Sign test results (p-value)

	Error Rate	R-Precision	Top 1 Precision	Top 3 Precision
Okapi vs. SVM	6.8e-10	7.64e-11	9.72e-11	4.18e-07
Okapi vs. Ranking SVM	1.33e-10	1.05e-08	1.31e-08	7.66e-07
Random vs. SVM	3.16e-12	9.05e-09	4.87e-06	1.31e-07
Random vs. Ranking SVM	1.06e-14	3.16e-10	1.71e-08	6.94e-08
SVM vs. Ranking SVM	0.295	0.200	0.311	1.000

1. Visio is a great product that too few people know about! We need to start driving internal use of Visio and show customers what Visio can do for them.
2. Visio is a drawing package designed to assist with the creation of a wide range of business diagrams, including flow-charts, process maps, database schema, building layouts, etc. Visio's approach is strongly graphical, allowing you to manipulate and format objects dropped onto the page.

Figure 6: Definition candidates for 'Visio'

From Table 4, we see that both Ranking SVM and SVM perform significantly better than Okapi and random ranking. The results indicate that our methods of using ordinal regression and classification for definition ranking are effective. We conducted a sign test on the improvements of Ranking SVM and SVM over Okapi. The results show that the improvements are significant (cf., Table 5).

It is not surprising that Okapi cannot work well for the task, because it is designed for search of relevant documents. In fact, relevance and goodness of definition are different notions. Figure 6 shows two examples of definition candidates for the term 'Visio'. Okapi ranks the first candidate ahead of the second candidate, because in the first candidate the query term 'Visio' repeats three times. However, the second candidate is a better definition than the first one. In contrast, Ranking SVM or SVM can rank the two candidates more appropriately, i.e., the second candidate is considered as a better definition than the first one.

We also conducted the sign test on improvement of Ranking SVM and SVM over random ranking and results show significant improvement too (cf., Table 5).

The performance of Ranking SVM is comparable with that of SVM. Our sign test results show that there is no significant difference between their ranking results in all measures. Both SVM and Ranking SVM have their own advantages. If there are more than three ordered categories (in our study, we happen to have three), we cannot easily simplify the problem as a classification problem. That is to say, ordinal regression is a more natural formalization for the task. On the other hand, although SVM has less representational power, it is usually more efficient to conduct model training for SVM than for Ranking SVM.

We conducted analysis on the erroneous results of Ranking SVM and SVM. The errors can be categorized as follows:

1. Negative effect of the adjective feature (good candidates are ranked to the bottom): 35%
2. Limitation of the features (indifferent or bad candidates are ranked on the top): 30%
3. Annotation error: 5%
4. Unknown reason: 30%

The adjective feature is a negative feature. That is the more adjectives a paragraph has the less likely the paragraph is a good definition. However, there are counter examples for which good definitions contain many adjectives. More sophisticated models are needed to address the problem.

Second, some paragraphs appear to be definitions if we only look at their first sentences. However, the entire paragraphs are not good definitions according to our specification (cf., the example paragraph in Figure 7 in which there is a topic change.). To cope with the problem, more useful features are needed.

SMTP is the protocol standard for transmitting electronic mail over the internet. Outlook 10 will have some changes in the way mail is sent, due to increases in ISP security and the unification of OMI and Corporate/Workgroup modes. SMTP is taken care of by a protocol handler that is controlled by other components of Outlook. It will take a group of emails that need to be sent out and transmit them one at a time. Success and errors are reported at the time of occurrence.

Figure 7: An example definition candidate for 'SMTP'

8.3 Ranking Definitional Paragraphs in TREC .gov Data

In the experiment, we tested whether generic models (SVM and Ranking SVM) can be constructed for ranking of definitions.

As training data, we used the same training data as in Section 8.2, which is from the Intranet data. As test data, we utilized the TREC .gov data set.

To create the test data, we employ the same method as described in Section 8.2. Table 6 shows the statistics of the test data. The data set contains 25 terms and 191 definition candidates. On average, each term has 2.68 (67/25) good definitions. The number is larger than that in Intranet data set. In Intranet data set, most definitions are about technical terms on products and product groups. In TREC .gov data, most definitions are about government sections and project names. The list of the 25 terms is given in Appendix.

Table 6: Statistics on TREC .gov paragraph data

Number of terms	25
Number of definition candidates	191
Number of good definitions	67
Number of indifferent definitions	76
Number of bad definitions	48

Table 7: Definitional paragraph ranking on TREC .gov data

	Error Rate	R-Precision	Top 1 Precision	Top 3 Precision
Okapi	0.4891	0.4267	0.4000	0.8000
SVM	0.2759	0.5747	**0.6400**	0.8400
Ranking SVM	**0.2466**	**0.5780**	**0.6400**	**0.9600**
Random Ranking	0.5100	0.3307	0.3200	0.7600

Table 7 shows that both Ranking SVM and SVM can achieve good results on the TREC .gov data set, although the models are trained in a different domain (Intranet). Both of them significantly outperform the baseline methods.

In Section 5.2.3, we have listed the features used in Ranking SVM and SVM. The features are domain independent. That it is why we can create a domain independent generic model.

8.4 Ranking Definitional Sentences in Intranet Data

In the experiment, we investigate the effectiveness of our approach when applied to ranking of definitional sentences.

We take the same term set and data as that in Section 8.2. For each term, we collect *<term, definition>* pairs and human annotators label them as good, indifferent or bad definitions. After that, terms without any good definition candidates are discarded. The final dataset contains 78 terms and 670 definition candidates. On average, each term has 2.01 (157/78) good definitions. The number is lower than that of paragraph candidates. It indicates that a sentence has a lower probability of being a good definition than a paragraph. Table 8 shows the statistics on the data.

Table 8: Statistics on Intranet sentence data

Number of terms	78
Number of definition candidates	670
Number of good definitions	157
Number of indifferent definitions	186
Number of bad definitions	327

In addition to the features used in Section 5.2.3, several new features are used for ranking of definitional sentences (e.g., position of the sentence in paragraph).

From Table 9, we see that both Ranking SVM and SVM perform significantly better than the baseline methods. (The results are also averaged over five trials in 5-fold cross validation.) The results suggest that our proposed methods based on Ranking SVM and SVM can work for definitional sentences ranking as well.

Table 9: Definitional sentence ranking on Intranet data

	Error Rate	R-Precision	Top 1 Precision	Top 3 Precision
Okapi	0.5986	0.2783	0.2564	0.5128
SVM	0.2022	0.6097	0.5972	0.8710
Ranking SVM	**0.1655**	**0.6769**	**0.7303**	**0.9365**
Random Ranking	0.4577	0.3693	0.3590	0.6795

9. CONCLUSIONS

In this paper, we have proposed to address the issue of searching for definitions by employing what we call definition ranking.

Under the setting, we have developed a new approach for conducting search of definitions. Specifically, we have proposed ranking definition candidates according to their goodness as definitions. Definition candidates are first extracted from documents using several simple rules. Next, the candidates are ranked using either Ranking SVM model or SVM model so that good definition candidates are on the top.

Experimental results indicate that our proposed methods perform significantly better than the baseline methods of using traditional IR and random ranking. The results also show that our proposed method works well for both paragraph level and sentence level definition ranking. They can also be easily adapted to different domains.

On the basis of the proposed methods, we have also developed an enterprise search system and put it into practical use.

The proposed methods are not limited to search of definitions. They can be used in search of other types of information as well.

10. ACKNOWLEDGMENTS

We are grateful to Prof. Yalou Huang of Nankai University and Prof. Jue Wang of Chinese Academy of Science for their encouragement and support. We express our gratitude to Cong Li for his preliminary work on the issue. We acknowledge Dmitriy Meyerzon for his making available the data in the experimentation. We thank Changning Huang, Ming Zhou, Hugo Zaragoza, and John Chen for their helpful comments to the paper.

11. REFERENCES

[1] E. Agichtein, S. Lawrence, and L. Gravano. Learning search engine specific query transformations for question answering. In Proc. of the 10th World Wide Web Conference, 2001.

[2] S. Blair-Goldensohn, K. R. McKeown, and A. H. Schlaikjer. Answering Definitional Questions: A Hybrid Approach (Chapter 4). In New Directions In Question Answering, Mark Maybury, ed. AAAI Press, 2003.

[3] S. Blair-Goldensohn, K. R. McKeown, and A. H. Schlaikjer. DefScriber: A Hybrid System for Definitional QA (Demo). In Proc. of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003.

[4] S. Blair-Goldensohn, K. R. McKeown, and A. H. Schlaikjer. A Hybrid Approach for QA Track Definitional Questions. In Proc. of the 12th Annual Text Retrieval Conference, 2003.

[5] H. Cui, M. Kan, and T. Chua. Unsupervised Learning of Soft Patterns for Definitional Question Answering. In Proc. of the 13th World Wide Web Conference, 2004.

[6] A. Echihabi, U. Hermjakob, E. Hovy, D. Marcu, E. Melz, and D. Ravichandran. Multiple-Engine Question Answering in TextMap. In Proc. of 12th Annual Text Retrieval Conference, 2003.

[7] S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, J. Williams, and J. Bensley. Answer Mining by Combining Extraction Techniques with Abductive Reasoning. In Proc. of the 12th Annual Text Retrieval Conference, 2003.

[8] R. Herbrich, T. Graepel, and K. Obermayer. Support Vector Learning for Ordinal Regression. In Proc. of the 9th International Conference on Artificial Neural Networks, 1999.

[9] T. Joachims. Optimizing Search Engines Using Clickthrough Data, Proc. of the 8th ACM Conference on Knowledge Discovery and Data Mining, 2002.

[10] J. Klavans and S. Muresan. DEFINDER: Rule-Based Methods for the Extraction of Medical Terminology and their

Associated Definitions from On-line Text. In Proc. of American Medical Informatics Association Symposium, 2000.

[11] G. Lakoff. Women, Fire, and Dangerous Things. What Categories Reveal about the Mind, Chicago University Press, Chicago, Ill, 1987.

[12] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. Sov. Phys. Dokl, 1966.

[13] B. Liu, C. Chin, and H. Ng. Mining Topic-Specific Concepts and Definitions on the Web. In Proc. of the 12th World Wide Web Conference, 2003.

[14] S. E. Robertson, S. Walker, M. M. HancockBeaulieu, M. Gatford, and A. Payne. Okapi at TREC-4. In Proc. of the 4th Annual Text Retrieval Conference, National Institute of Standards and Technology, Special Publication 500-236, 1995.

[15] G. Salton and M. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

[16] M. H. Sarner and S. Carberry. A new strategy for providing definitions in task-oriented dialogues. In Proc. of the 12th Conference on Computational Linguistics, 1988.

[17] Q. Tan, X. Chai, W. Ng, and D. L. Lee. Applying Co-training to Clickthrough Data for Search Engine Adaptation. In Proc. of the 9th International Conference on Database Systems for Advanced Applications, 2004.

[18] V. N. Vapnik. The Nature of Statistical Learning Theory. Springer, 1995.

[19] E. Voorhees. Evaluating Answers to Definition Questions. In Proc. of HLT-NAACL, 2003.

[20] E. Voorhees. Overview of the TREC 2003 Question Answering Track, In Proc. of the 12th Annual Text Retrieval Conference, 2003.

[21] J. Xu, A. Licuanan, and R. Weischedel. TREC 2003 QA at BBN: Answering Definitional Questions. In Proc. of the 12th Annual Text Retrieval Conference, 2003.

[22] E. Xun, C. Huang, and M. Zhou. A Unified Statistical Model for the Identification of English BaseNP. In Proc. of the 38th Annual Meeting of the Association for Computational Linguistics, 2000.

[23] H. Yang, H. Cui, M.-Y. Kan, M. Maslennikov, L. Qiu, and T.-S. Chua. QUALIFIER in TREC-12 QA Main Task. In Proc. of the 12th Annual Text Retrieval Conference, 2003

[24] http://www.google.com/help/features.html#definitions.

APPENDIX

List of terms in TREC .gov data

1. AIDS
2. ATLAS
3. Advanced Spaceborne Thermal Emission and Reflection Radiometer
4. Agency for Toxic Substances and Disease Registry
5. Alcohol
6. Breast Cancer
7. Department of Health and Human Services
8. Diabetes
9. FBI
10. FDA
11. FOIA
12. FTC
13. HRSA
14. IRS
15. Intermountain Precipitation Experiment
16. Java
17. MAP
18. MTBE
19. NIST
20. NOAA Weather Radio
21. NSF
22. NTIA
23. Science Bowl
24. Sexual harassment
25. U.S. Geological Survey