May 2006, Vol.21, No.3, pp.439–449 J. Comput. Sci. & Technol.

A Supervised Learning Approach to Search of Definitions*

Jun Xu[1] (徐　君), Yun-Bo Cao[2] (曹云波), Hang Li[2] (李　航), Min Zhao[3] (赵　珉), and Ya-Lou Huang[1] (黄亚楼)

[1] *College of Software, Nankai University, Tianjin 300071, P.R. China*

[2] *Microsoft Research Asia, Beijing 100080, P.R. China*

[3] *Institute of Automation, Chinese Academy of Sciences, Beijing 100080, P.R. China*

E-mail: nkxj@yahoo.com.cn; {yucao, hangli}@microsoft.com; m.zhao@mail.ia.ac.cn; yellow@nankai.edu.cn

Received May 29, 2005; revised February 14, 2006.

Abstract This paper addresses the issue of search of definitions. Specifically, for a given term, we are to find out its definition candidates and rank the candidates according to their likelihood of being good definitions. This is in contrast to the traditional methods of either generating a single combined definition or outputting all retrieved definitions. Definition ranking is essential for tasks. A specification for judging the goodness of a definition is given. In the specification, a definition is categorized into one of the three levels: good definition, indifferent definition, or bad definition. Methods of performing definition ranking are also proposed in this paper, which formalize the problem as either classification or ordinal regression. We employ SVM (Support Vector Machines) as the classification model and Ranking SVM as the ordinal regression model respectively, and thus they rank definition candidates according to their likelihood of being good definitions. Features for constructing the SVM and Ranking SVM models are defined, which represent the characteristics of terms, definition candidate, and their relationship. Experimental results indicate that the use of SVM and Ranking SVM can significantly outperform the baseline methods such as heuristic rules, the conventional information retrieval—Okapi, or SVM regression. This is true when both the answers are paragraphs and 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.

Keywords definition search, text mining, web mining, web search

1 Introduction

Definitions describe the meanings of terms and thus belong to the type of frequently accessed information. It is helpful if we develop a system that can automatically find definitions of terms from documents on the web.

Traditional information retrieval is designed to search for relevant documents to the submitted query (e.g., [1]), and thus is not suitable for performing the task.

TREC (Text REtrieval Conference) formalizes the problem as that of definitional question answering[2,3]. Given the question of "what is X" or "who is X", a system extracts answers from multiple documents and combines the extracted answers into a single unified answer[4−8]. Question answering is ideal, as means of helping people to find definitional information. However, it might be difficult to materialize it in practice. Usually definitional sentences of a term 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[9,10]. Most of the methods resort to human-defined rules for definition extraction.

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 their likelihood of being good definitions.

Definition ranking is essential for the task. Suppose that we have a number of definition candidates such as those shown in Fig.1. By ranking of definition we mean sorting definition candidates in descending order of their degrees of goodness as definitions. In this way, users can easily get the definition information they look for.

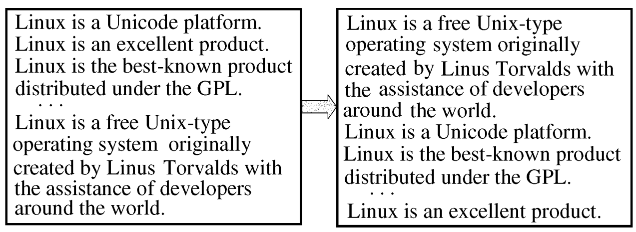

Fig.1. Definition ranking.

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. A specification for judging whether a definition is "good", "bad", or "indifferent" is proposed. SVM (Support Vector Machines) and Ranking SVM models are employed as our classification and ordinal regression models respectively. We

Regular Paper

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

also develop a set of features used in the SVM and Ranking SVM models. Definition ranking is performed 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 starting 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.

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. Other experimental findings are that the trained 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 definition. Section 5 explains our approaches to definition ranking. Section 6 describes our approach to definition search. Section 7 reports our experimental results. Section 8 summarizes our work in the paper.

2 Related Work

2.1 Automatically Discovering Definitions

Google offers a feature of definition search ①. When users type "define: ⟨*term*⟩" in the search box, the search engine returns glossaries containing the definitions of the ⟨*term*⟩. This feature relies on the fact that there are many glossary pages available on the Internet. We are not clear how Google collects the glossary pages, but it seems that all 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[2,3].

Systems have been developed for performing the question answering task in TREC. In TREC 2003, most of the systems[4–8] 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[6] 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.

We note that there is also a step of ranking in the previous TREC systems such as that in [6]. However, the ranking in the systems is based on the answer candidates' importance and similarities to the query term. In our work, definition candidates are ranked on the basis of their goodness as definitions. We think, therefore, that the ranking techniques developed in TREC may not be suitable for the task in this paper.

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

For instance, DEFINDER[9] 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 then combines the extracted results of the two modules.

Liu *et al.* proposed a method of mining topic-specific knowledge on the web[10]. 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 definitional question answering or definition discovery, see [11–20].

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.*[21] proposed an algorithm for this task. In their paper, learning of the preference between objects is formulated as a classification of pairs of objects and is solved using the principle of structural risk minimization. Joachims[22] proposed learning a ranking function for search as ordinal regression using clickthrough data. He employs what he calls the Ranking SVM model for ordinal regression. For other related work on ordinal regression, see [23–30].

①http://www.google.com/help/features.html

3 Definition Search

First, let us describe the task of "definition search" more precisely. 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 one after another according to the likelihood of each candidate being a good definition and output them.

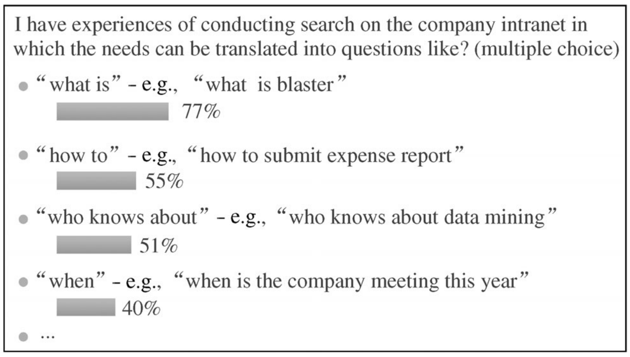

Fig.2. Survey on experiences of search in an IT company.

Without loss of generality, in this paper we only consider the 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, on a company intranet. We conducted a survey in an IT company in which we asked the employees what kind of searches they had ever performed on their company intranet. Fig.2 shows the result of one question. We see that 77% of the people had experiences of searching for "what is" questions, i.e., definitions.

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

We tried to collect glossaries on an intranet of a company and found that there were only a few glossaries available. We collected all the web pages containing at least one of the keywords "glossary", "gloss", "dictionary", "definition", or "define" and manually checked whether they were glossary pages. From about 1,000,000 web pages in total, we were only able to find about 50 glossary pages containing about 1,000 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 Fig.3).

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, background color, image placement and so on.

Fig.3. Definitions of HTML from different perspectives.

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 for definition of the concept[31].

Furthermore, the qualities of definitions extracted from different documents can vary. Usually, there are many descriptions which cannot 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.

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.

Fig.4. Example definition candidates for Linux.

Fig.4 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 of 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 are available. This is because such data are not always available, particularly when it is on an intranet.

In the text mining methods described in Subsection 2.1, the extracted definitions are treated uniformly and thus are not ranked. As we have discussed, however, definitions should be sorted according to 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 of 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 them the specification in this paper. It is indispensable for the 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 a 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 Fig.4 are examples of a good definition.

A bad definition describes neither 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 Fig.4 are examples of a bad definition.

An indifferent definition is the one that between good and bad definitions. Sentences 4–7 in Fig.4 are examples.

5 Definition Ranking

In definition ranking, we extract, from the entire collection of documents, $\langle$*term, definition, score*$\rangle$ 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. It means that we filter out all unlikely candidates. Second, we calculate the score of each candidate collected for definition, using an SVM or Ranking SVM. As a result, we obtain triples of $\langle$*term, definition, score*$\rangle$. Third, we find similar definitions by 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. In general, it 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. And 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 first sentences in the paragraphs with a Base NP (base noun phrase) parser and identify $\langle$*term*$\rangle$s using the following rules. (For the definition of Base NP, see for example [32].)

1) $\langle$*term*$\rangle$ is the first Base NP of the first sentence.

2) Two Base NPs separated by "of" or "for" in the first sentence are combined as $\langle$*term*$\rangle$. For example, "Perl for ISAPI" is the term from the first sentence "Perl for ISAPI is a plug-in designed to run Perl scripts$\cdots$"

In this way, we can identify not only the single-word $\langle$*term*$\rangle$s, but also more complex multi-word $\langle$*term*$\rangle$s.

Next, we pull out definition candidates with the following patterns,

1) $\langle$*term*$\rangle$ is a|an|the $*$

2) $\langle$*term*$\rangle$, $*$, a|an|the $*$

3) $\langle$*term*$\rangle$ is one of $*$

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

The step of collecting definition candidates is similar to the method of definition extraction employed in [10]. The uses of other sets of rules for candidate selection are also possible. However, they are not essential if the approach is to conduct ranking of definition candidates. As mentioned above, we can skip this step or reinforce it by using a smaller or larger number of rules.

5.2 Ranking Definition Candidates

At this step, we determine the goodness of a candidate as a definition. The goodness of a definition candidate is determined by the nature of the paragraph and is independent of the term itself. Thus, the ranking on the basis of goodness of definition differs from the ranking on the basis of relevance to query in traditional information retrieval.

We take a statistical machine learning approach to addressing 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 $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, we construct a model that can minimize error in prediction of y given $\boldsymbol{x}$ (generalization error). Here $\boldsymbol{x}_i \in X$ and $y_i \in \{good,\ indifferent,\ bad\}$ represent a definition candidate and a label, respectively. When applied to a new instance $\boldsymbol{x}$, 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 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, for there is an order between the three categories.

We employ Ranking SVM[22] as the model of ordinal regression. Given an instance $\boldsymbol{x}$ (definition candidate), Ranking SVM assigns a score to it based on

$$\boldsymbol{U}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}, \qquad (1)$$

where $\boldsymbol{w}$ represents a vector of weights. The higher the value of $\boldsymbol{U}(x)$ is, the better the instance $\boldsymbol{x}$ as a definition is. In ordinal regression, the values of $\boldsymbol{U}(\boldsymbol{x})$ 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 [22]. 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)[33] as the model of classification. Given an instance $\boldsymbol{x}$ (definition candidate), SVM assigns a score to it based on

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b, \qquad (2)$$

where $\boldsymbol{w}$ denotes a vector of weights and b denotes a intercept. The higher the value of $f(\boldsymbol{x})$, the better the instance $\boldsymbol{x}$ as a definition. In the classification, the sign of $f(\boldsymbol{x})$ is used. If it is positive, then $\boldsymbol{x}$ 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 [33]. 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 SVM and Ranking 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*

SVM and Ranking SVM utilize the same set of binary or real valued features. Table 1 shows the list of the features.

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*⟩.

We discovered the feature set with the following processes. We first constructed a set of training data. Next, we looked at the data and created an initial set of features. Then, we conducted an experiment to evaluate the performance of using the feature set. We added or removed features to tune the models until we could not further improve the performances. Finally, we got the feature set as that shown in Table 1.

Features 2, 3, 5, and 6 are used to characterize the targeted term of a definition candidate. Features 4, 9, 11, and 12 are used to characterize the entire definition candidate. There are also features 1, 7 and 8 for representing the relationship with the targeted term.

There are positive features like Feature 1. That is, if the term appears at the beginning of the paragraph, then it is likely that the paragraph is a definition on the term. There are also negative features like Feature 4. If words like "she", "he", or "said" occur in the paragraph, it is likely that the paragraph is not a (good) definition. (We can tell whether a feature is positive or negative from the linear SVM and Ranking SVM. After building up SVM or Ranking SVM models, we check the weight of each feature[33]. If the weight is positive, then the feature is a positive feature. Otherwise, it is a negative feature.)

We also utilize bag-of-words features. High frequency words appear 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. Some keywords work as positive features while others as negative features.

5.3 Remove 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[34]. If two definition candidates are too similar, we remove the one which is ranked lower. Currently, we set the similarity threshold as 0.8.

6 Search for Definitions

All the data necessary for definition search are stored in a database table in advance. The data are 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.

In search for 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. For example, given the query term "Linux", we retrieve the ranked list of the definition candidates as those in Table 2. The candidates are ranked by their scores.

7 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.

Currently there is no standard data set for evaluation of definition search methods, as far as we know. We created two data sets by ourselves. The two data sets are from a corpus of a company intranet and from the TREC .gov corpus. They are referred to as "Intranet" and "TREC".

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

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.

7.1 Baseline and Measure for Evaluation

Most definitional QA systems aim to generate a single answer for a given question. There is no QA system available for performing the task of definition search defined in the paper.

We made use of three baseline methods.

First, we used Okapi[35]. Given a query term, it returns a list of paragraphs or sentences ranked only on the basis of relevance to the query term.

We also used random ranking of definition candidates. This can be viewed as an approximation of the existing methods of definition extraction.

Finally, we used SVM regression in which the ranks in the labeled data were mapped into real numbers.

We made use of three measures for evaluation of definition ranking. They are "error rate of preference pairs" (cf., [21, 22]), R-precision (precision of R highest ranked candidates, where R is the number of "good" definitions), and Top N precision (percentage of terms whose top N ranked candidates contain "good" definitions. $N = 1$ or 3). (3), (4) and (5) give the details.

$$\mathit{Error\ rate} = \frac{|\mathit{mistakenly\ predicted\ preference\ pairs}|}{|\mathit{all\ preference\ pairs}|}. \tag{3}$$

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

where R is the number of good definitions form $term_i$.

$$\mathit{R\text{-}precision} = \frac{\sum_{i=1}^{T} \mathit{R\text{-}precision}\ (\mathit{term}_i)}{T}, \tag{4}$$

where T is the number of terms in data set.

Top N precision $=$

$$\frac{|\textit{terms whose top N ranked candidates contain "good"}|}{|\textit{all terms in data set}|}. \quad (5)$$

It is difficult to evaluate the results in terms of recall, because the data sets are large and it is too costly to manually find all the definitions from the entire data sets.

7.2 Ranking Definitional Paragraphs in Intranet Data

In this experiment, we considered a paragraph as an instance for definition search. First, we extracted all the definition candidates, i.e., $\langle term,\ definition\rangle$ pairs from the Intranet corpus. 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 those 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 1,366 candidates. On average, each term has 2.4 (225/95) good definitions. Table 3 shows statistics on the data. We tested the effectiveness of ranking with both Ranking SVM and SVM using the data set.

Table 3. Statistics on Intranet Paragraph Data

Number of terms	95
Number of definition candidates	1,366
Number of good definitions	225
Number of indifferent definitions	470
Number of bad definitions	671

We conducted 5-fold cross validation. We randomly divided terms and their corresponding definition candidates into five even subsets. In each trial, one of the five subsets was used as the test set and the other four subsets were put together to form a training set. We trained the ranking model on the training set and tested its performance on the test set. It was repeated five times and the results reported in Table 4 are those averaged over the five trials.

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
Random Ranking	0.4363	0.3224	0.3474	0.6316
SVM Regression	0.3636	0.3335	0.3811	0.7144
SVM	0.3284	0.4658	0.4324	0.8351
Ranking SVM	**0.2712**	**0.5180**	**0.5502**	**0.8868**

In the experiment, for SVM regression, we mapped the three ranking levels "bad", "indifferent", and "good" into the real numbers of 1, 2, and 3, respectively. Other mapping methods were also tried. The results were not good, however.

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

For Ranking SVM, we tried to use both query dependent paragraphs and query independent paragraphs as data to train the model. We found that the performances of the two methods were similar. In the following experiments, query independent data were used.

From Table 4, we see that both Ranking SVM and SVM outperform Okapi, random ranking, and SVM regression significantly. The results indicate that our methods of using ordinal regression and classification for definition ranking are effective. We conducted sign test and t-test on the improvements of Ranking SVM and SVM over Okapi, random ranking, and SVM regression. The results show that the improvements are significant in all measures (cf., Table 5 and Table 6).

It is not surprising that Okapi cannot work well for the task, because it is designed for search for relevant documents. In fact, relevance and goodness of definition are different notions. Fig.5 shows two examples of definition candidates for the term "Visio". Okapi ranks the first candidate ahead of the second candidate, because

Table 5. Sign Test Results (p-Value)

	Error rate	R-precision	Top 1 Precision	Top 3 Precision
Okapi vs. SVM	6.8×10^{-10}	7.64×10^{-11}	9.72×10^{-11}	4.18×10^{-7}
Okapi vs. Ranking SVM	1.33×10^{-10}	1.05×10^{-8}	1.31×10^{-8}	7.66×10^{-7}
Random vs. SVM	3.16×10^{-12}	9.05×10^{-9}	4.87×10^{-6}	1.31×10^{-7}
Random vs. Ranking SVM	1.06×10^{-14}	3.16×10^{-10}	1.71×10^{-8}	6.94×10^{-8}
Regression vs. Ranking SVM	0.0006	0.0043	0.0243	0.0135
Regression vs. SVM	0.0094	0.0075	0.0043	0.0078
SVM vs. Ranking SVM	0.2950	0.2000	0.3110	1.000

Table 6. t-Test Results (p-Value)

	Error rate	R-precision	Top 1 Precision	Top 3 Precision
Okapi vs. SVM	6.9614×10^{-11}	9.86×10^{-9}	8.44×10^{-10}	1.71×10^{-7}
Okapi vs. Ranking SVM	1.6425×10^{-11}	4.81×10^{-11}	1.71×10^{-12}	8.85×10^{-8}
Random vs. SVM	1.6615×10^{-7}	7.21×10^{-8}	9.25×10^{-7}	3.25×10^{-8}
Random vs. Ranking SVM	7.4962×10^{-11}	2.12×10^{-10}	1.92×10^{-9}	1.63×10^{-8}
Regression vs. SVM	0.0200	0.0023	0.0170	0.0225
Regression vs. Ranking SVM	0.0061	0.0026	0.0078	0.0662
SVM vs. Ranking SVM	0.0967	0.0707	0.1194	0.3538

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.

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 flowcharts, process maps, database schema, building layouts, etc. Visio's approach is strongly graphical, allowing you to manipulate and format objects dropped onto the page.

Fig.5. Definition candidates for Visio.

Random ranking is an approximation of existing methods of definition extraction, which rely on patterns and rules. It does nothing for ranking definition candidate. Thus it cannot work well for the task either.

SVM regression requires mapping the rank labels into real numbers. The model appears to be an unsuitable model for the task.

We also noticed that random ranking performs a litter better than Okapi (cf. Table 4), which indicates ranking based on relevance between query and document is not suitable for the task of ranking definitions.

The performance of Ranking SVM is comparable with that of SVM. Our sign test and t-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 an SVM model.

We investigated the effectiveness of individual features. The results are reported in Fig.6. The i-th bar denotes the performance (R-precision) drop when the i-th feature is removed from Ranking SVM. The descriptions of features can be found in Table 1. From Fig.6, we can see the performance drops significantly when feature 1, 2, 9, or 10 is removed. We also see that bag-of-words features (feature 12) are important for the ranking task.

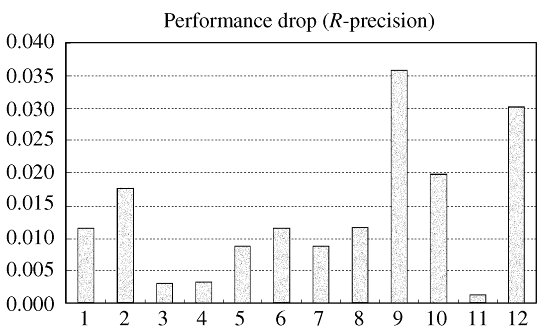

Fig.6. Performance drop when features were removed.

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 and features are needed to address the problem.

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 Fig.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, for the 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.

Fig.7. Example definition candidate for SMTP.

7.3 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. Constructing generic models means that we can train the ranking models in one domain and apply the models to other domains. Many efforts for annotating data can be saved if domain adaptation is easy.

As for training data, we used the same training data as in Subsection 7.2, which is from the Intranet corpus. We utilized the TREC corpus as test data. Notice that we did not need conduct cross validation here.

To create the test data, we employ the same method as described in Subsection 7.2. Table 7 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 the Intranet data set. In the Intranet data set, most definitions are about technical terms of products and product groups. In the TREC .gov data, most definitions are about government sections and project names.

Table 8 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).

Table 7. 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 8. 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
Random Ranking	0.5100	0.3307	0.3200	0.7600
SVM Regression	0.4217	0.3713	0.3800	0.8000
SVM	0.2759	0.5747	**0.6400**	0.8400
Ranking SVM	**0.2466**	**0.5780**	**0.6400**	**0.9600**

In Subsection 5.2.3, we have listed the features used in Ranking SVM and SVM. The features are domain independent. We think that is why we can create a domain independent generic model.

7.4 Ranking Definitional Sentences in Intranet Data

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

We take the same term set and data as that in Subsection 7.2. 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 contains less information than a paragraph and has a lower probability of being a good definition. Table 9 shows the statistics on the data.

Table 9. 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 Subsection 5.2.3, several new features as shown in Table 10 are used for the ranking of definitional sentences. These features characterize the context (paragraph) information. Some features, such as 1 and 6, are positive features. It seems that people usually define a term at the beginning of a paragraph and then refer to the term. Some features like feature 2 are negative features. People seldom give a definition at the end of a paragraph.

From the result shown in Table 11, we see that both Ranking SVM and SVM outperform the baseline methods significantly. (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 be able to work for definitional sentences ranking as well.

Table 10. Features Added in Ranking Definitional Sentences

1. The candidate is the first sentence in paragraph.
2. The candidate is the last sentence in paragraph.
3. Number of previous sentences in paragraph.
4. Number of following sentence in paragraph.
5. ⟨*term*⟩ occurs at preceding sentences in paragraph.
6. ⟨*term*⟩ occurs at posterior sentences in paragraph.

Table 11. 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
Random Ranking	0.4577	0.3693	0.3590	0.6795
SVM Regression	0.2108	0.4561	0.4576	0.7588
SVM	0.2022	0.6097	0.5972	0.8710
Ranking SVM	**0.1655**	**0.6769**	**0.7303**	**0.9365**

8 Conclusion

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 to conducting search for 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 outperform the baseline methods significantly using traditional IR, random ranking, and SVM regression. 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.

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

Acknowledgements We are grateful to Prof. Jue Wang of Chinese Academy of Sciences for his encouragement. We express our gratitude to Cong Li for his preliminary work on the issue. We acknowledge Dmitriy Meyerzon for his making available data in the experimentation. We thank Chang-Ning Huang, Ming Zhou, Hugo Zaragoza, and John Chenfor their helpful comments on the paper.

References

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

[2] Voorhees E. Evaluating answers to definition questions. In *Proc. Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics Annual Meeting,* Edmonton, Canada, 2003, pp.109–111.

[3] Voorhees E. Overview of the TREC 2003 question answering track. In *Proc. 12th Text Retrieval Conference*, Washington, 2003, pp.54–68.

[4] Blair-Goldensohn S, McKeown K R, Schlaikjer A H. A hybrid approach for QA track definitional questions. In *Proc. 12th Text Retrieval Conference*, Washington, 2003, pp.185–192.

[5] Harabagiu S, Moldovan D *et al.* Answer mining by combining extraction techniques with abductive reasoning. In *Proc. 12th Text Retrieval Conference*, Washington, 2003, pp.375–382.

[6] Xu J, Licuanan A, Weischedel R. TREC 2003 QA at BBN: Answering definitional questions. In *Proc. 12th Text Retrieval Conference*, Washington, 2003, pp.98–106.

[7] Echihabi A, Hermjakob U, Hovy E *et al.* Multiple-engine question answering in TextMap. In *Proc. 12th Text Retrieval Conference*, Washington, 2003, pp.772–781.

[8] Yang H, Cui H, Kan M Y *et al.* QUALIFIER in TREC-12 QA main task. In *Proc. 12th Text Retrieval Conference*, Washington, 2003, pp.480–488.

[9] Klavans J, Muresan S. DEFINDER: Rule-based methods for the extraction of medical terminology and their associated definitions from on-line text. In *Proc. American Medical Informatics Association Symposium*, Los Angeles, CA, USA, 2000, pp.201–202.

[10] Liu B, Chin C, Ng H. Mining topic-specific concepts and definitions on the web. In *Proc. 12th Int. Conf. World Wide Web*, Budapest, Hungary, May 20–24, 2003, pp.251–260.

[11] Agichtein E, Lawrence S, Gravano L. Learning search engine specific query transformations for question answering. In *Proc. 10th Int. Conf. World Wide Web*, Hong Kong, May 1–5, 2001, pp.169–178.

[12] Blair-Goldensohn S, McKeown K R, Schlaikjer A H. Answering Definitional Questions: A Hybrid Approach (Chapter 5). New Directions in Question Answering, Maybury M (ed.), AAAI Press, 2004.

[13] Blair-Goldensohn S, McKeown K R, Schlaikjer A H. DefScriber: A hybrid system for definitional QA. In *Proc. 26th Annual Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Toronto, Canada, July 28–August 1, 2003, pp.462–462.

[14] Cui H, Kan M, Chua T. Unsupervised learning of soft patterns for definitional question answering. In *Proc. 13th Int. Conf. World Wide Web*, New York, USA, May 17–20, 2004, pp.90–99.

[15] Kwok K L. Experiments with a component theory of probabilistic information retrieval based on single terms as document components. In *ACM Trans. Information Systems* (*TOIS*), 1990, 8(4): 363–386.

[16] Kwok K L. A network approach to probabilistic information retrieval. In *ACM Trans. Information Systems* (*TOIS*), 1995, 13(3): pp.325–354.

[17] Radev D R, Libner K, Fan W. Getting answers to natural language questions on the web. *Journal of the American Society for Information Science and Technology* (*JASIST*), 2002, 53(5): 359–364.

[18] Radev D R, Fan W, Qi H, Wu H, Grewal A. Probabilistic question answering on the web. *Journal of the American Society for Information Science and Technology* (*JASIST*), 2005, 56(6): 571–583.

[19] Sarner M H, Carberry S. A new strategy for providing definitions in task-oriented dialogues. In *Proc. 12th Conf. Computational Linguistics*, Budapest, Hungry, August 22–27, 1988, pp.567–572.

[20] Xu J, Weischedel R, Licuanan A. Evaluation of an extraction-based approach to answering definitional questions. In *Proc. 27th Annual Int. Conf. Research and Development in Information Retrieval*, Sheffield, UK, 2004, pp.418–424.

[21] Herbrich R, Graepel T, Obermayer K. Support vector learning for ordinal regression. In *Proc. 9th Int. Conf. Artificial Neural Networks*, Edinburgh, UK, 1999, pp.97–102.

[22] Joachims T. Optimizing search engines using clickthrough data. In *Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 23–26, 2002, pp.133–142.

[23] Crammer K, Singer Y. Pranking with Ranking. Advances in Neural Information Processing Systems 14, Dietterich T G, Becker S, Ghahramani Z (eds.), Cambridge, MA: MIT Press, 2002.

[24] Fan W, Gordon M D, Pathak P. Genetic programming based discovery of ranking functions for effective web search. *Journal of Management Information Systems* (*JMIS*), 2005, 21(4): 37–56.

[25] Fürnkranz J, Hüllermeier E. Pairwise preference learning and ranking. In *Proc. 14th European Conf. Machine Learning* (*ECML2003*), Cavtat-Dubrovnik, Croatia, Sept. 22–26, 2003, pp.145–156.

[26] Frank E, Hall M. A simple approach to ordinal classification. In *Proc. 12th European Conf. Machine Learning* (*ECML2001*), Freiburg, Germany, Sept. 5–7, 2001, pp.145–156.

[27] Shashua A, Levin A. Taxonomy of Large Margin Principle Algorithms for Ordinal Regression Problems. Advances in Neural Information Processing Systems 15, Becker S, Thrun S, Obermayer K (eds.), Cambridge, MA: MIT Press, 2003, pp.937–944.

[28] Tan Q, Chai X, Ng W, Lee D L. Applying co-training to clickthrough data for search engine adaptation. In *Proc. the 9th Int. Conf. Database Systems for Advanced Applications*, Lecture Notes in Computer Science 2973, Jeju Island, Korea, March 17–19, 2004, pp.519–532.

[29] Harrington E F. Online ranking/collaborative filtering using the perceptron algorithm. In *Proc. 12th Int. Conf. Machine Learning* (*ICML2003*), Washington, USA, August 21–24, 2003, pp.250–257.

[30] Kramer S, Widmer G, Pfahringer B, Groeve M. Prediction of ordinal classes using regression trees. In *Proc. 12th Int. Symp. Methodologies for Intelligent Systems*, Charlotte, NC, USA, Oct. 11–14, 2000, pp.426–434.

[31] Lakoff G. Women, Fire, and Dangerous Things. Chicago: Chicago University Press, Ill, 1987.

[32] Xun E, Huang C, Zhou M. A unified statistical model for the identification of English BaseNP. In *Proc. 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, 2000, pp.547–554.

[33] Vapnik V N. The Nature of Statistical Learning Theory. Springer-Verlag, 1995.

[34] Levenshtein V I. Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys.*, 1966, 10(8): 707–710.

[35] Robertson S E, Walker S, HancockBeaulieu M M *et al.* Okapi at TREC-4. In *Proc. 4th Text Retrieval Conference*, National Institute of Standards and Technology, Special Publication 500-236, 1995, pp.73–96.

Jun Xu is a Ph.D. candidate at College of Information Science and Technology, Nankai University, China. He received his B.S. degree in computer science and technology, from Nankai University in June 2001. His main research interests include text mining, information retrieval, and natural language processing.

Yun-Bo Cao is a researcher at Microsoft Research Asia. He received his M.S. degree in computer science, from Peking University in June 1997. His main research interests include statistical learning, natural language processing, data mining, and information retrieval.

Hang Li is a researcher at Microsoft Research Asia. He is also an adjunct professor of Xian Jiaotong University and Nankai University. He joined Microsoft Research in June 2001. Prior to that, he worked at the Research Laboratories of NEC Corporation. He obtained the B.S. degree in electrical engineering from Kyoto University in 1988 and the M.S. degree in computer science from Kyoto University in 1990. He earned his Ph.D. degree in computer science from the University of Tokyo in 1998. His research interests include statistical learning, natural language processing, data mining, and information retrieval.

Min Zhao is currently a researcher at NEC Laboratories China. She received her Ph.D. degree from Institute of Automation, Chinese Academy of Sciences. Her main research interests include rough set, data mining, and information retrieval.

Ya-Lou Huang is a professor at Nankai University. He received his Ph.D. degree in control theory and control engineering from Nankai University in 1993. His research interests include intelligent robot, intelligent information processing, and data mining.