A Comparative Study on Sentence Retrieval for Definitional Question Answering

Hang Cui Min-Yen Kan Tat-Seng Chua Jing Xiao

Department of Computer Science
School of Computing
National University of Singapore
{cuihang, kanmy, chuats, xiaojing}@comp.nus.edu.sg

ABSTRACT

Most definitional question answering (QA) systems integrate statistical ranking using Web and WordNet as external resources and pattern matching to retrieve relevant sentences for further processing. We examine the impact of using these two common resources in answering definition questions by varying the use of WordNet and two types of Web resources in statistical ranking, and definition pattern modules in a typical definitional QA system. In particular, we show that an appropriate combination of Web-reinforced ranking and soft matching pattern produces an unsupervised system that outperforms the best system at TREC-12 by 6%.

Keywords

Definitional question answering, pattern matching, soft patterns, Web knowledge, WordNet, comparative evaluation.

1. INTRODUCTION

Definition questions, *i.e.,* questions like "What is *TB*?" or "Who is *Aaron Copland*?" have drawn much attention recently [16]. A typical definitional question answering (QA) system extracts definition sentences that contain the most descriptive information about the search term from multiple documents and summarizes the sentences into definitions.

At the recent TREC-12 conference, an entire evaluation devoted to definitional QA was conducted [17]. Although the top performing definitional QA systems fielded at TREC [4, 19, 20] differ on specifics, they concur on the general architecture for definition sentence retrieval which includes: a) a statistical ranking component that leverages evidence about the search term from both the corpus and external resources, such as WordNet [5] and the Web; and b) a definition pattern matching component. The statistical ranking component identifies significant terms that bear central information on the search term to locate relevant documents or sentences. In order to find more accurate terms to describe the search term, this component often employs a variety of external resources, *e.g.,* WordNet and the Web, to find the basic definition of the search term. Since not all sentences that contain significant words are definitional, definition pattern matching sifts through the sentences to find matches with certain syntactic styles that are common to definitions, such as appositives and copulas. Although these techniques are widely used in definitional QA, we are not aware of any systematic study of definition sentence retrieval using component evaluations. What is the impact of tweaking IR performance with the use of external resources and different pattern matchers, and how does this affect overall QA performance?

We seek to answer this question by examining definitional QA in a standard system while varying the configurations of two commonly used constituents for sentence retrieval, namely, statistical ranking using external knowledge, and definition pattern matching. Specifically, in the statistical ranking component, we use WordNet and the Web as external resources of definitions. We further differentiate Web knowledge into two types: general Web search results (using search engines) and information from specific definitional Web sites (e.g., online encyclopedia). Pattern matching is an idiosyncrasy of definition sentence retrieval compared to other passage retrieval methods, requiring deeper syntactic analysis. Although most successful systems employ manually constructed definition patterns, we use machine learning to generate patterns automatically, as we believe manually constructed patterns consume considerable labor and are thus hard to obtain and maintain. We employ an adaptive rule induction system in information extraction (IE) for definition pattern learning. In addition to the automatic generalization of hard matching rules[1], which requires strict matching (*i.e.*, matching slot by slot), we examine soft pattern generation and matching. Cui *et al*. [3] showed that soft matching patterns are effective in extracting definition sentences, which may vary greatly in lexical and syntactic forms. Their evaluations, while helpful, are not complete. We complete their comparison of hard matching and soft matching patterns by including experiments using machine generalized hard matching rules from the IE system.

Our key findings include: (1) Specific Web knowledge gleaned from definitional Web sites greatly improves the performance of definitional QA while results from general Web searches yield only moderate improvement. (2) Rule induction algorithms can be applied to definition pattern learning, and may provide a wider coverage of definition patterns than manually constructed rules. (3) Soft matching patterns outperform hard matching rules that are manually constructed or machine learned. (4) Judicious use of Web knowledge boosts unsupervised learning of soft patterns as a result of improved statistical ranking of sentences.

We show that an appropriate combination of statistical ranking using Web knowledge and soft matching patterns produces a totally unsupervised system that outperforms the best system at TREC-12 by 6%.

2. RELATED WORK

Early TREC systems such as the FALCON system [6] employed simple, manually constructed definition patterns to extract proper phrases. The recent TREC-12 systems employ more complicated techniques. Xu *et al*. [19] and Echihabi *et al*. [4] integrated

[1] We use "pattern" and "rule" interchangeably in this paper.

manually constructed definition patterns with centroid-based statistical ranking (see Section 5). In addition to using corpus statistics, they assigned credit to those sentences that contain frequently occurring terms in definitions or biographies from certain Web sites (*e.g.,* biography.com). They also include WordNet [5] as a kind of external knowledge. In this paper, we study the effectiveness of using WordNet and the Web in statistical ranking, respectively. Further examples of state-of-the-art definitional QA systems include [20], which utilizes task-general Web knowledge – snippets from Google search – to supplement statistical ranking; and [1] which utilizes supervised learned rules to identify specific syntactic components. Recent work by Hildebrandt *et.al.* [7] merges answers from pre-complied database extracted by manually constructed patterns, existing dictionaries and the corpus to generate definitions. The commonality in the above systems serves as the basis for our design of the consensus definitional QA architecture.

In domain-specific definitional QA, Schiffman *et al.* [13] proposed to combine data-driven statistical methods and machine learned rules to produce biographical summaries for people, *i.e.,* answering "who is" questions. They based the summaries mainly on appositives and relative clauses. Liu *et al.* [8] proposed mining topic-specific definitions for scientific subjects from the Web, and relied on a set of manually constructed rules to find definition sentences.

Lexio-syntactic patterns have been utilized in question answering systems. Soubottin and Soubottin [15] manually complied patterns for answering factoid questions. To date, most existing definitional QA systems employ manually constructed definition patterns while ignoring a variety of machine learning techniques for generalizing pattern rules automatically. Ravichandran and Hovy [11] proposed unsupervised technique to learn surface patterns for question answering. However, their system is tuned mainly for factoid questions instead of definition questions. The patterns they found cover only a small portion of definition patterns. Much work has been done in obtaining lexico-syntactic patterns for information extraction [9]. To examine the effectiveness of such techniques in creating definition patterns, we study an adaptive supervised rule induction system [18] in generalizing definition patterns from training instances. Another kind of pattern generation technique is the soft pattern matching proposed by Cui *et al.* [3]. Soft pattern matching performs instance-based learning, and thus does not generalize training instances into specific rules. Soft patterns keep all positive training instances and are represented as a virtual vector. Soft matching patterns match against a test sentence using a probabilistic framework instead of using regular expressions for hard matching rules.

In contrast to existing research, our work performs separate evaluations for two common influential components within a definitional QA system. We also show the interaction of the two components in the context of unsupervised pattern learning.

3. DEFINITIONAL QA SYSTEM ARCHITECTURE

We use a standard definitional QA system which conforms to the consensus architecture for our comparative study. Its architecture is illustrated in Figure 1. Given a definition question, the system proceeds to construct definitions in three main steps: (1) document and passage retrieval to get relevant sentences about the search term, (2) sentence retrieval, and (3) sentence selection to choose non-redundant definition sentences from the results of sentence retrieval to form the definition. We employ a standard information retrieval system with anaphora resolution for Step (1), and a variation of Maximal Marginal Relevance (MMR) [2] to accomplish Step (3). The sentence retrieval module, which is the object of this study, integrates statistical ranking (2a) and pattern matching (2b) to produce a list of definition sentences. While it is possible to place Step (2b) before Step (2a), we employ such a common structure adopted in many of the current definitional QA systems [1, 20]. In this paper, we vary the configurations of the sentence retrieval module while using fixed settings for the other two modules.

The statistical ranking component (Step 2a) leverages statistical evidence from both the corpus and external resources. We will present it alongside WordNet and different Web knowledge resources in Section 5. The definition pattern module (Step 2b) checks whether a pattern from its pattern repository applies to the test sentence. If a definition pattern matches, the weight of the sentence from the statistical ranking module is modified accordingly. The definition patterns can come from different sources – manually-created rules, generalized rules from a rule induction algorithm, and soft patterns. We detail our experiments on varying the pattern module in Section 6.

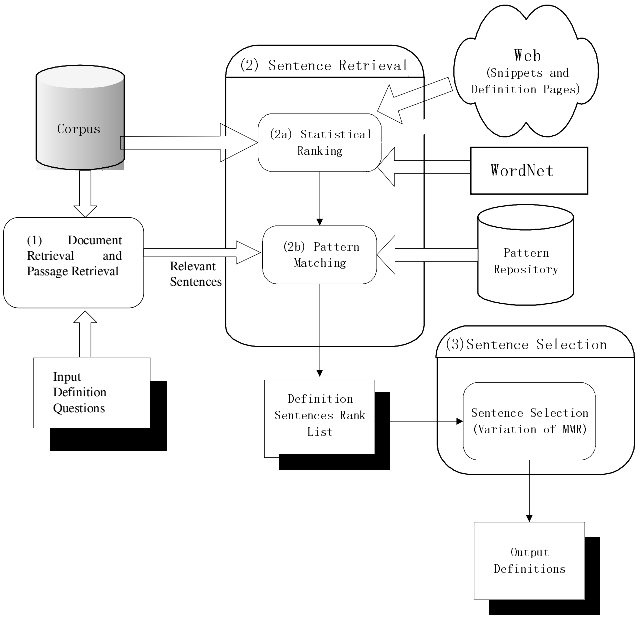
Web
(Snippets and Definition Pages)
(2) Sentence Retrieval
Corpus
(2a) Statistical Ranking
WordNet
(1) Document Retrieval and Passage Retrieval
Relevant Sentences
(2b) Pattern Matching
Pattern Repository
(3)Sentence Selection
Input Definition Questions
Definition Sentences Rank List
Sentence Selection (Variation of MMR)
Output Definitions

Figure 1. Architecture of the Definitional QA System

4. EVALUATION SETUP

We use the community standard TREC-12 definitional QA data set [17]. The TREC QA corpus[2] comprises over one million news articles, which are accompanied by 50 definition questions and answers in the form of answer nuggets. Among the questions, there are 30 questions about people (*e.g., Aaron Copland*), 10 about organizations (*e.g., ETA*), and 10 about other terms (*e.g., quasar*).

In order to provide additional training data for pattern learning beyond the TREC corpus, an auxiliary set of Web

[2] The AQUAINT Corpus of English News Text. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T31

documents are collected based on questions from Lycos. This Lycos training set comprises 26 questions on people and other terms most frequently searched for in Lycos (http://50.lycos.com). For each Lycos question, we use Google's site search to obtain up to 200 news articles from each of eight prominent news sites (*e.g.,* CNN and BBC). The body text of each news page is extracted. We asked seven subjects to label all definition sentences. The labeled sentences are processed into 596 positive and 15,442 negative training instances. Note that the Lycos training set is used only for pattern learning and not as external knowledge in the statistical ranking component.

We adopt the same evaluation metrics as in the TREC-12 task. For each question, TREC gives a list of essential nuggets[3] and acceptable nuggets for answering the question. An individual definition question is scored using nugget recall (NR) and an approximation to nugget precision (NP) based on answer length. These scores are combined using the F_5 ($\beta = 5$) measure, where recall is five times as important as precision. We attempt to simulate the evaluation process done by TREC by asking a subject to examine the results for official answer nuggets provided by TREC. The NP, NR and F_5 measures are calculated based on that subject's judgments.

We set the length of definitions to ten sentences for people and seven sentences for other terms according to our preliminary experiments over the development data set.

5. STATISTICAL RANKING USING EXTERNAL KNOWLEDGE

We examine the role of external knowledge in influencing the statistical ranking step because most definitional QA systems employ external knowledge at this stage. The idea is that words that appear in definitions from WordNet or the Web, or co-occur frequently with the search target in retrieved snippets are likely to express specific definition facets of the search term. These words should be stressed in statistical ranking.

As illustrated in Figure 1, the statistical weighting of sentences (Step 2a) is performed using statistics from both the corpus and external resources. When ranking sentences with corpus word statistics, we employ the centroid-based ranking method, which has been used in other definitional QA systems (*e.g.,* [1, 19]) and document summarization [10]. We select a set of *centroid words* (excluding stop words) which co-occur frequently with the search target in the input sentences. To select centroid words, we use mutual information to measure the centroid weight of a word *w* as follows:

$$Weight_{centroid}(w) = \frac{\log(Co(w, sch\_term)+1)}{\log(sf(w)+1)+\log(sf(sch\_term)+1)} \times idf(w) \quad (1)$$

where *Co(w, sch_term)* denotes the number of sentences where *w* co-occurs with the search term *sch_term*, and *sf(w)* gives the number of sentences containing the word *w*. We also use the inverse document frequency of *w*, *idf(w)*[4], as a measurement of the global importance of the word. Words whose centroid weight exceeds the average plus a standard deviation are selected as centroid words. We form centroid words into a centroid vector, which is then used to rank input sentences by their cosine similarity with the vector.

The weighting of centroid words can be improved by using external knowledge. In the first configuration, we search the question targets in WordNet and obtain the gloss of these words as definitions. WordNet covers 20 out of 50 questions in our evaluation set. In other configurations, we use two types of Web resources in our evaluations: task-independent information, provided in the form of Google snippets; and task-specific information, in the form of definitions from the online encyclopedia site Wikipedia (www.wikipedia.com) and the online biography site Biography.com (www.biography.com). We attempt to obtain 200 snippets from Google for each search term. For task-specific information, we retrieve the whole definition text from Wikipedia and Biography.com for questions that have Web site entries. While there are many other Web sites that can be used as sources for definitions, we take these two web sites as representative samples to examine their impact on the performance. Wikipedia gives descriptions of people and terms, covering 34 of our 50 questions. Biography.com provides definitions for people only, covering 23 of the 30 questions about people. Combined, the task-specific Web sites cover 42 questions and contribute significantly less words than the task-general Google snippets do. We augment the weight of those words that also occur in the text retrieved by WordNet or Web resources:

$$Weight_{CW}(w) = \begin{cases} Weight_{Centroid}(w) \times (1+\log(SF(w)+1)) & \textit{if w occurs in general Web resouces} \\ Weight_{Centroid}(w) \times (1+\gamma) & \textit{if w occurs in specific definitions} \end{cases} \quad (2)$$

where *Weight_{Centroid}(w)* denotes the centroid weight of the word *w* obtained by Equation (1). *SF(w)* gives the number of snippets that contain the word *w* while γ is a constant factor. We try different γ values (from 0.2 to 1.0) to optimize our system and set it to 0.6 based on our preliminary experiments.

5.1 Evaluations

We use centroid-based ranking as the baseline in the comparisons. We fix the pattern matching component with a set of manually constructed rules as listed in Appendix 1. These rules are extracted from [20], a system which tied for second place in the TREC-12 definitional QA task. We note here that without incorporating any definition patterns, using only the centroid-based method achieves a F_5 value of 42.38%. This means that the manually constructed rules bring an improvement of 9.21% over the statistical method. We vary the use of WordNet, task-general and task-specific Web knowledge and assess their impact on the baseline system. In cases where both general and specific Web resources cover the same search term, we use the specific resources. We do not include a configuration that includes both Web knowledge and the use of WordNet. This is due to that WordNet provides only short definitions to the question terms and the definitions are mostly covered by the task-specific Web knowledge. We plan to examine their combination in the future work. The results are shown in Table 1.

[3] See (Voorhees, 2003) for the definitions of answer nugget, NR, NP and F_5.

[4] We use the statistics from Web Term Document Frequency and Rank site (http://elib.cs.berkeley.edu/docfreq/) to approximate words' IDF.

Table 1. Impact of Using Web Knowledge

Web Knowledge configuration (Specific and General)	NR	NP	F_5 Measure (% improvement)
1. Centroid + Manually Constructed Rules (Baseline)	51.00	19.53	46.69
2. Baseline + WordNet	56.13	19.72	50.88 (+8.97%)
3. Baseline + Task-general Web	51.45	20.69	47.27 (+1.24%)
4. Baseline + Task-specific Web	58.05	21.71	53.37 (+14.32%)
5. Baseline + Task-general and Task-specific Web	**58.55**	**21.59**	**53.86 (+15.37%)**

The following observations are derived from Table 1:

First, specific Web resources are more useful than general Web resources in helping find more definition sentences. The use of task-general search engine snippets results in only minute improvement while the use of task-specific Web resources brings an significant improvement of 14.32% ($p<0.05$) over the baseline. If we only count the 42 questions which are covered by the two specific Web sites, the percentage improvement over the baseline is 19.40%. We attribute the improvement to the fact that specific definitional Web sites provide succinct and accurate definition to the search target such that evidence from specific Web knowledge can precisely reflect definitional facets of the search target. WordNet is another task-specific resource for finding definitions since we use the retrieved gloss of words in the experiments. However, WordNet performs worse than specific definitional Web sites because WordNet covers only a small percentage of questions and provides only short definitions for the question targets. In contrast, task-general Web knowledge, obtained from general search engines, returns relevant information about the search term, in which definitional descriptions account for only a small part. As such, general Web knowledge can provide only very limited help.

Second, using the task-general Web in addition to the task-specific Web yields a small additional performance gain. Using only specific Web knowledge runs the risk that some questions may not be covered. Additional knowledge gleaned from general Web resources can compensate for the possibility that information on some questions may be lacking at specific definitional Web sites.

Third, when we remove manually constructed rules from the final combination, *i.e.,* we use only statistical ranking that combines corpus statistics with all Web knowledge, the F_5 measure drops from 53.86% to 50.61% (not shown in Table 1) with NP from 21.59% to 20.51% and NR from 58.55% to 56.25%, respectively. This result suggests that even with the direct help of definitions from external resources, appropriate definition patterns still play an important role in identifying definition sentences within the corpus. Definition patterns help improve both the precision and recall of definition sentence retrieval. The reason is that definition patterns can not only filter out those statistically highly-ranked sentences that are not definitional, but also bring those definition sentences that are written in certain styles for definitions but are not statistically significant into the answer set.

6. PATTERN EXTRACTION

Definition patterns, such as "X, a …" and "X is a …" differentiate definition sentences from other sentences short-listed by statistical ranking. As discussed before, manually constructed patterns are widely used and have achieved acceptable results. However, manually constructed patterns rely heavily on the knowledge of the developer. As definitions can be expressed in many forms, with some being quite obscure (*e.g.*, "X (also called …)"), an exhaustive list seems difficult to obtain by manual means. As such, we believe that manually constructed rules are likely to be limited by their poor adaptability and low coverage.

In order to compare the use of manually constructed rules against automatically generated patterns, we explore different techniques to generate pattern rules from training data. We first use a state-of-the-art rule induction algorithm borrowed from information extraction. This approach generalizes hard pattern rules over supervised training data and matches the rules using regular expressions. We also examine soft matching patterns to address the flexibility in expressing definitions. We will experimentally demonstrate the effectiveness of both machine learned hard patterns and soft patterns, as compared to manually constructed ones.

Moreover, we study the interaction of statistical ranking coupled with Web knowledge and pattern matching in the context of unsupervised learning of patterns. Specifically, we show the impact of combining Web knowledge in unsupervised learning of soft patterns in another set of evaluations.

6.1 Preprocessing

To conduct automatic pattern learning, we first obtain a set of labeled definition sentences as training samples. In order to learn generic patterns, we perform two steps to prepare the training instances. First, words specific only to the search targets are replaced with general tags in order to ensure the generality of the learned patterns. Second, we crop the text windows surrounding the search target as training instances.

In the first step, the labeled definition sentences are processed with part-of-speech (POS) tagging and chunking by a natural language tagger and chunker[5]. We then perform selective substitution of certain lexical items by their syntactic classes. The substitution replaces words that are specific to the search term with more general tags. For instance, centroid words determined by the statistical ranking module are replaced by their POS classes as the words are specific to the search term and do not help in forming general definition patterns. The query terms for the search target are substituted and concatenated to <SCH_TERM>. For instance, given a definition sentence for the search term "Iqra":

The channel Iqra is owned by the Arab Radio and Television company and is the brainchild of the Saudi millionaire, Saleh Kamel.

After substitution, the sentence is transformed into a token sequence comprising POS tags, generalized tags (marked by a final $), words and punctuation:

DT$ NN <SCH_TERM> BE$ owned by DT$ NP and BE$ DT$ brainchild of NP .

[5] We used NLprocessor, a commercial parser from Infogistics Ltd. http://www.infogistics.com/.

In the second step, we generate training pattern instances by cropping the contextual windows around <SCH_TERM> to *w* tokens to the left and the right. We note here that definition sentences are identified mainly by adjacent words and punctuation. The value of *w* depends on the rule generation methods. As rule induction systems that learn hard rules can automatically prune slots with little support, we use a large window size of 5 for hard pattern generation. For soft pattern generation, we set *w* to 2; this setting has been shown to be optimal in our cross-validation experiments. We refer to the cropped fragments as *training instances*. The training instance from the above sentence is (*w*=2):

DT$ NN <SCH_TERM> BE$ owned

After pre-processing, the labeled definition sentences are turned into a group of training instances, from which we conduct rule generation of both hard and soft patterns.

6.2 Hard Pattern Rule Induction by GRID

Machine induced rules are widely used for information extraction [9]. To adapt a rule induction system for information extraction to definition pattern learning, we apply GRID [18], a state-of-the-art supervised rule induction algorithm. We select GRID for two reasons. First, unlike other rule induction algorithms that start with seed rules [12] or randomly selected instances [14], GRID uses corpus-wide distribution statistics to start the rule induction process. This is likely to fit well with the diversity in definition patterns. Second, GRID utilizes both tokens and coarse-grained tags (*e.g.*, POS and phrase level tags) in learning rules. The rules learned by GRID are represented as regular expressions. We employ GRID over the labeled definition sentences from the auxiliary Lycos training set. An excerpt of the generated rules is shown in Figure 2.

1. <SCH_TERM> , DT NN
2. <SCH_TERM> , DT NNP
3. <SCH_TERM> , who won
4. <SCH_TERM> , (known | listed) as
5. who BE <SCH_TERM> 's
6. <SCH_TERM> BE DT NN

Figure 2. Sample rules generated by GRID

For rules that are manually constructed or generalized by machine learning techniques, hard matching is performed to match test sentences to the rules. Hard matching rules may be problematic when the slot of an instance could take different forms in a sentence. For instance, the rule "*<SCH_TERM>, DT NN*" generated by GRID would match the sentence "*Goth*, a subculture ……". However, it would fail to match the sentence "*Goth* – a subculture ……", although the two differ by only a single punctuation slot. This problem can be partially solved by soft matching patterns.

6.3 Soft Pattern Extraction and Matching

As definition sentences often exhibit variations in vocabulary and syntax, the use of hard pattern rules seems too rigid: it is not tolerant to noise in training data, and often cannot recognize definition patterns which are not explicitly accounted for in the training set. To overcome this problem, Cui *et al.* [3] proposed a soft matching technique which computes the degree of match between the test sentences and the training instances using a probabilistic model.

Given a set of training instances, a virtual vector representing the soft definition pattern *Pa* is generated by aligning the training instances according to the positions of <SCH_TERM>:

$<Slot_{-w}, \dots Slot_{-2}, Slot_{-1}, SCH\_TERM\ , Slot_{1}, Slot_{2}, \dots Slot_{w} : Pa>$

where $Slot_i$ contains a vector of tokens with their probabilities of occurrence derived from the training instances.

The test sentences are first preprocessed in a manner similar to the preprocessing of labeled definition sentences. Using the same window size *w*, the token fragment *S* surrounding the <SCH_TERM> is retrieved:

$<token_{-w}, \dots, token_{-2}, token_{-1}, SCH\_TERM, token_{1}, token_{2}, \dots token_{w} : S>$

The matching degree of the test sentence to the generalized definition patterns is measured by the similarity between the vector *S* and the virtual soft pattern vector *Pa*. The matching degree is calculated in two parts. The first part calculates the degree of similarity between individual slots, assuming independence between slots. The second part examines sequence fidelity by adopting a bigram model[6]. The slot and the sequence similarity weights are linearly combined to determine the overall pattern matching weight of a test sentence.

6.4 Evaluation: The Impact of Definition Patterns

We have two hypotheses concerning the use of definition patterns: (a) Manually-constructed patterns ought to be of high precision but low recall, due to the difficulty in enumerating an exhaustive specification of definition patterns. Machine-learned patterns may do better at recall by learning from large-scale training data. (b) Soft matching patterns should outperform hard matching systems as they can capture the flexibility in a definition structure. To validate these hypotheses, we conduct a series of experiments using the TREC corpus.

We maintain the baseline configuration described in Section 5 as the basis for comparison. In the second configuration, we replace the manually constructed rules by a set of 100 hard rules in regular expressions generalized by the GRID algorithm over the Lycos training set. This set of hard rules is denoted as "GRID Lycos HP". The third test explores the use of soft patterns derived from all positive instances from the Lycos training set. The resulting group of soft patterns is denoted as "Lycos SP".

To combine statistical weighting with pattern matching, we apply different strategies to hard matching rules and soft patterns: As the match is binary for manually constructed rules and generalized hard rules by GRID, the weight of any sentence that matches a rule has its score multiplied by a constant factor *g*, which is set to 2; again, this is the optimum setting that we have ascertained in our validation experiments by varying the setting from 1.2 to 3. When applying soft pattern matching, the sentences are re-ranked by the linear combination of statistical and pattern matching weights. We weight evidence from pattern matching higher because we believe that patterns are better able to identify definition sentences.

[6] See [3] for details of the soft matching method.

Table 2. Comparison of definition patterns

Use of Patterns	NR	NP	F_5 Measure (% improvement)
1. Centroid + Manually Constructed Rules (Baseline)	51.00	19.53	46.69
2. Centroid + GRID Lycos HP	53.61	22.16	49.75 (+6.56%)
3. Centroid + Lycos SP	**63.00**	**24.26**	**55.98 (+19.92%)**

The evaluation results are presented in Table 2. We make the following observations:

First, machine learned patterns outperform the manually constructed ones. As many of the TREC-12 top-performing systems use manually constructed patterns, they are likely to benefit from automatic pattern learning. We see an improvement of 6.56% in the F_5 measure over the manually constructed rules when using the generalized hard patterns generated by GRID. When we apply the soft matching patterns over the supervised Lycos pattern instances, the improvement rises significantly to 19.92%. This validates our hypothesis that manually constructed rules are often limited in recall. We expect a larger performance gain with more training instances.

Second, soft patterns significantly outperform hard patterns. Applying soft patterns over the supervised Lycos pattern instances, the system performs 12.53% better than when using GRID generalized hard rules. This improvement is statistically significant (*$p<0.01$*). We conjecture that soft patterns can better capture infrequent definition patterns as they use all positive instances in the construction of a flexible probabilistic model. Hard-matching rule induction systems may ignore such infrequent data. In addition, strict slot-by-slot matching may miss some positive instances that exhibit minor variations in expressions, which are common to definitions. Soft patterns thus provide a mechanism to overcome these problems.

6.5 The Impact of Web Knowledge in Unsupervised Soft Pattern Learning

In addition to performing separate component evaluations, we also study the interaction of Web knowledge and pattern matching. Specifically, we examine the role of Web knowledge in unsupervised soft pattern learning as a case study.

We adopt the group pseudo-relevance feedback technique (GPRF) proposed in Cui, *et al.* [3] to accomplish automatic labeling of training instances. For each question, we first rank all input sentences using the statistical ranking component (Step 2a in Figure 1). The top *k* (*k*=10 in our evaluations according to the original setting) ranked sentences for every question are treated as labeled definition sentences. Soft pattern generation is conducted on the blindly labeled sentences of *all* questions. Cui *et al.* [3] employed only corpus statistics in statistical ranking and obtained only 33% of the automatically labeled sentences as definitional. We have noted in Section 5 that the use of Web resources helps in finding more appropriate weights for centroid-based ranking. So, the question naturally arises: Can we integrate soft matching patterns and Web resources in an unsupervised learning system to gain further improvement?

In our final evaluations, we use centroid-based ranking and soft patterns learned from unsupervised labeled definition sentences determined by GPRF as the baseline. We apply combinations of task-general and task-specific Web resources to boost the retrieval performance of centroid-based weighting as we did in Section 5. We also include an experiment that leverages more offline learned patterns, in the form of additional supervised soft patterns learned over the Lycos training set. We present the results in Table 3.

Table 3. Integration of soft patterns and Web resources

Configuration	NR	NP	F_5 Measure (% improvement)
1. Centroid + GPRF SP (Baseline)	60.11	22.19	53.91
2. Baseline + General Web	61.89	22.09	55.56 (+3.06%)
3. Baseline + Specific Web	65.08	24.56	58.74 (+8.96%)
4. Baseline + General Web + Specific Web	65.24	23.49	58.76 (+9.00%)
5. Baseline + Lycos SP + General Web + Specific Web	**65.48**	**23.36**	**58.96 (+9.36%)**

We see that the use of Web knowledge boosts the overall performance of unsupervised soft pattern matching. We can draw the following observations from Table 3:

First, the performance of GPRF-based unsupervised labeling is comparable to that of supervised learning, which is presented in Table 2. This shows that soft matching makes our method robust to noisy training data. Although some of the top sentences by statistical ranking for individual search terms may not be definitional, the use of GPRF on the batch of questions helps to minimize the effect of idiosyncratic errors from sentences that come from single questions.

Second, we re-affirm that task-specific Web resources make the most improvement in definitional QA. Task-specific Web resources bring a significant improvement of 8.96% (*$p<0.02$*) over the baseline while using general Web knowledge can boost the performance only by a modest 3.06%. Combining Web resources with soft pattern matching affects the final performance in two ways: (a) Web resources improve the statistical ranking process by providing more precise and redundant information on the definition of the search term. (b) As a result of the improved statistical ranking, the GPRF process is able to bring more actual definition sentences to the top list so that more accurate pattern instances are obtained. Therefore, the accuracy of soft patterns is improved accordingly.

Third, when both offline learned Lycos soft patterns and online learned soft patterns through GPRF are employed along with all Web knowledge, we get the highest performance of 0.5896 in F_5 measure. This is 6% higher than the best TREC-12 system, which has an F_5 score of 0.555 [19]. This test shows that integrating more offline trained patterns to unsupervised learned patterns helps in definitional QA as we are able to derive a large soft pattern repository learned over many different corpora.

7. LIMITATIONS OF EVALUATIONS

We have shown a series of extrinsic experiments for studying definition sentence retrieval. There are two main limitations in our evaluation framework.

First, as our focus has been definition sentence retrieval, we have deliberately neglected the impact of the other two modules in a typical definitional QA system, namely document/passage retrieval (Step 1 in Figure 1) and sentence selection (Step 3 in Figure 1). While we may have evaluated an end-to-end system, we have used uniform settings in these two modules for all experiments. Actually, many factors within these two modules may potentially affect overall performance. For instance, anaphora resolution in passage retrieval is crucial in finding definition sentences where the search target is not explicitly stated, which is common in writings. In addition, we have empirically set the length of final definitions in the sentence selection module. Experimental results show that due to the definition of F_5 measure (recall as five times important as precision), longer definition length may boost overall performance.

Second, we have not optimized the parameters of each subsystem adopted in our comparative study. Instead, we have either used the default settings of the systems or empirically set values based on our preliminary development experiments. To make the whole framework more robust, it is imperative for us to find a systematic way to automatically adjust important parameters, such as the factor for weighting Web knowledge in statistical ranking and that for weighting patterns in sentence scores.

Third, as argued in [7], there are inconsistencies even in TREC official evaluations. Although we try our very best to simulate the evaluation process of TREC and stick to the official answer nuggets provided, it is inevitable for the assessor to vary in the judgments across evaluations. The main purpose of this paper is to provide first-hand results on the impacts of different components for definitional QA. We expect to have more stable evaluation results by using more questions in the evaluations.

8. CONCLUSION

Recent definitional QA systems integrate statistical ranking that utilizes external knowledge and pattern matching to extract definition sentences. Within this architecture, we studied the effects of employing different external resources to enhance statistical ranking and different methods for pattern generation and matching.

We have quantified the performance gain by using WordNet and different sources of Web knowledge. Specifically, we have shown that task-specific Web knowledge can greatly impact the performance in comparison to the mediocre improvement manifested by task-general Web resources. We therefore recommend that future definitional QA systems should select resources well-suited to their tasks and weight such external information with care. A more systematic use of external knowledge should also be explored in future work.

We have shown that machine learning methods for pattern generation outperform manually constructed patterns used by most definitional QA systems. Definition patterns can be derived by different pattern generation techniques.

In addition to reaffirming the effectiveness of soft matching patterns over hard matching ones, we also tested the idea of unsupervised learning of soft patterns by combining Web knowledge in the first round of statistical ranking. We have shown that incorporating more explicit evidence in statistical ranking augments the quality in the automatic labeling of training instances, and that in turn improves unsupervised soft pattern generation, and ultimately enhances overall system performance.

9. ACKNOWLEDGEMENT

The authors would like to thank Alexia Leong for proofreading this paper. The first author is supported by Singapore Millennium Foundation Scholarship (ref no. 2003-SMS-0230). The authors are also grateful to anonymous reviews for their valuable comments for improving this paper.

10. REFERENCES

[1] S. Blair-Goldensohn, K.R. McKeown and A. Hazen Schlaikjer, *A Hybrid Approach for QA Track Definitional Questions*, The Twelfth Text REtrieval Conference (TREC 2003) Notebook, pp. 336-343, 2003.

[2] J. Carbonell and J. Goldstein, *The use of MMR, diversity-based reranking for reordering documents and producing summaries*, in Proceedings of the 21st ACM-SIGIR International Conference on Research and Development in Information Retrieval, Melbourne, Australia, 1998, pp. 335-336.

[3] H. Cui, M.–Y. Kan and T.-S. Chua, *Unsupervised Learning of Soft Patterns for Generating Definitions from Online News*, Proceedings of the Thirteenth World Wide Web conference (WWW 2004), New York, May 17-22, 2004, pp. 90-99.

[4] A. Echihabi, U. Hermjakob, E. Hovy, D. Marcu, E. Melz and D. Ravichandran, *Multiple-Engine Question Answering in TextMap*, The Twelfth Text REtrieval Conference (TREC 2003) Notebook, pp. 713-722, 2003.

[5] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[6] S. Harabagiu, D. Moldovan, R. Mihalcea M. Pasca, R. Bunescu, M. Surdeanu, R. G irju, V. Rus, and P. Morarescu, *Falcon: Boosting knowledge for answer engines*, Proc. Of Ninth Text Retrieval Conference (TREC 9), pp. 479-488, 2000.

[7] W.Hildebrandt, B.Katz and J.Lin, *Answering definition questions using multiple knowledge sources,* Proceedings of HLT/NAACL 2004, Boston, MA, May 2 – 7, 2004, pp. 49-56.

[8] B.Liu, C-W. Chin and H-T. Ng, 2003, *Mining Topic Specific Concepts and Definitions on the Web*, In Proceeding of International Conference on World Wide Web, 2003, Budapest, Hungary, pp. 251-260.

[9] I. Muslea. *Extraction patterns for information extraction tasks: A survey*. In AAAI-99 Workshop on Machine Learning for Information Extraction, 1999, pp.1-6.

[10] D. Radev, H. Jing and M. Budzikowska, *Centroid based summarization of multiple documents*, in ANLP/NAACL '00 Workshop on Automatic Summarization (Seattle, WA, April 2000) pp. 21-29.

[11] D. Ravichandran and E. Hovy, *Learning Surface Text Patterns for a Question Answering System*, In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 41-47.

[12] E. Riloff. *Automatically generating extraction patterns from untagged text*. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, pages 1044-1049, 1996.

[13] B. Schiffman, I. Mani, and K. J. Concepcion. *Producing biographical summaries: Combining linguistic knowledge with corpus statistics*. In Proceedings European Association for Computational Linguistics, 2001.

[14] S.G. Soderland, *Learning Text Analysis Rules for Domain-Specific Natural Language Processing*. PhD thesis, University of Massachusetts Amherst, 1997.

[15] M. M. Soubbotin and S. M. Soubbotin, *Patterns of Potential Answer Expressions as Clues to the Right Answers,* Tenth Text REtrieval Conference (TREC-10), Gaithersburg, MD. November 13-16, 2001.

[16] E.M.Voorhees, *Overview of the TREC 2001 question answering track*, Proceedings of the Eleventh Text REtrieval Conference (TREC 2001), 2001.

[17] E.M.Voorhees, *Overview of the TREC 2003 question answering track*, The Twelfth Text REtrieval Conference (TREC 2003) Notebook, 2003.

[18] J. Xiao, T.S. Chua and J. Liu, *A Global Rule Induction Approach to Information Extraction*, Proceedings of 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03), Sacramento, California, USA, November 03 - 05, 2003, pp.530-536.

[19] J. Xu, A. Licuanan and R. Weischedel, *TREC 2003 QA at BBN: Answering Definitional Questions*, The Twelfth Text REtrieval Conference (TREC 2003) Notebook, pp. 28-35, 2003.

[20] Hui Yang, Hang Cui, Min-Yen Kan, Mstatislav Maslennikov, Long Qiu and Tat-Seng Chua, *QUALIFIER in TREC 12 QA Main Task*, The Twelfth Text REtrieval Conference (TREC 2003) Notebook, pp. 54-63, 2003.

APPENDIX

Appendix 1. List of manually constructed rules

ID	Regular expressions of rules
1	<SCH_TERM> (who | which | that)* (is | are) (called | known as)*
2	<SCH_TERM> , (a | an | the)
3	<SCH_TERM> (is | are) (a | an | the)
4	<SCH_TERM> , or
5	<SCH_TERM> (- | :)
6	<SCH_TERM> (is | are) (used to | referred to | employed to | defined as | described as)
7	" (.+) " by <SCH_TERM>
8	(called | known as | referred to) <SCH_TERM>

Legend:

| - Any one of the elements within the round brackets

** - Optional field*

(.+) – Any character