Unsupervised Learning of Soft Patterns for Generating Definitions from Online News

Hang Cui Min-Yen Kan Tat-Seng Chua
Department of Computer Science,
School of Computing,
National University of Singapore,
Singapore, 117543.
{cuihang, kanmy, chuats}@comp.nus.edu.sg

ABSTRACT
Breaking news often contains timely definitions and descriptions of current terms, organizations and personalities. We utilize such web sources to construct definitions for such terms. Previous work has identified definitions using hand-crafted rules or supervised learning that constructs rigid, hard text patterns. In contrast, we demonstrate a new approach that uses flexible, soft matching patterns to characterize definition sentences. Our soft patterns are able to effectively accommodate the diversity of definition sentence structure exhibited in news. We use pseudo-relevance feedback to automatically label sentences for use in soft pattern generation. The application of our unsupervised method significantly improves baseline systems on both the standardized TREC corpus as well as crawled online news articles by 27% and 30%, respectively, in terms of F measure. When applied to a state-of-art definition generation system recently fielded in the TREC 2003 definitional question answering task, it improves the performance by 14%.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Search and Retrieval - Relevance feedback, Retrieval models, Search process, Selection process.

General Terms
Algorithms, Experimentation.

Keywords
Definition generation, definitional question answering, soft patterns, pseudo-relevance feedback, unsupervised learning.

1. INTRODUCTION
With the rapid expansion and ubiquity of the Web, the public often learns about breaking stories and developments in online news. New terms and personalities, such as Enron, Clay Aiken and SARS, which are of great interest to the public, are often described in such media. These emerging and constant changing terms and personalities can often be found only in breaking news web sites, but not in authoritative sources of definitions, such as dictionaries or encyclopedias. Traditional web searching is only part of the solution: on such a topic, a search can retrieve relevant web pages from news sites, but cannot filter these pages down to a single, coherent definition. To synthesize a complete definition of any such entity requires the identification and collation of definition sentences across relevant articles.

We focus on identifying definition sentences from relevant news articles for recent terms for which structured knowledge bases (*i.e.*, WordNet, Internet-accessible glossary, or machine readable dictionary) have no definition. A definition sentence contains descriptive information that can be included in an extended definition of the term. Such an "extended definition" answers not only "what/who is X?", but also "what/who is X like?" [7]. To create a final coherent definition, sentence editing and re-ordering may be employed, but are beyond the scope of this paper.

Most approaches applicable to our problem formulation use some form of pattern matching to identify definition sentences. [5] employed a simple method which defines several manually-constructed definition patterns to extract definition phrases. [17] and [2] summarized syntactic components, such as appositives and predicates, using generic rules learned from annotated corpus. [9] proposed to mine topic-specific definitions using hand-crafted rules to find definition sentences in web pages. These approaches have two shortcomings that we have identified and address in this work:

1. Pattern inflexibility: Whether using corpus-based learning techniques or manually creating patterns, to our knowledge all previous systems create hard-coded rules that require strict matching (*i.e.*, matching slot by slot). Although such hard patterns are widely used in information extraction [10], we feel that definition sentences display more variation and syntactic flexibility that may not be captured by hard patterns. In contrast, we propose a novel method which utilizes soft-matching patterns. Soft patterns take each slot as a vector of words and syntactic classes with their distributions, rather than generalizing specific instances to induce rules. This allows us to match test instances against the patterns using a probabilistic similarity measure. The learned soft patterns are used to judge whether sentences are definitional.
2. Manual labor: Manually constructed definition patterns are also limited by the ability of the developer to exhaustively enumerate all applicable patterns, known to be a difficult problem. Lack of pattern coverage results directly in low recall. Supervised learning can compensate for this weakness to some extent, but is limited by the availability of annotated corpora, which requires intensive labor and hinders the portability to other domains. In contrast, our soft patterns can be learned in an unsupervised manner. While our approach to soft pattern learning is robust to noise, we apply pseudo-relevance feedback (PRF) to boost

Copyright is held by the author/owner(s).
WWW 2004, May 17–20, 2004, New York, New York, USA.
ACM 1-58113-844-X/04/0005.

the quality of the initial retrieval set of definition sentences. By applying PRF before soft pattern induction, we can skip the laborious tasks of corpus annotation and pattern construction.

To demonstrate the effectiveness of our techniques, we have implemented a fully-functional definition generation system which constructs definitions for terms or person names by extracting definition sentences from relevant input news articles. That system can also be treated as a definitional QA system [20] because it provides direct answers to definition questions. We carry out a series of extrinsic evaluations to assess the performance of soft pattern matching and pseudo relevance feedback, and to assess their portability to the web. Our experiments used the TREC corpus to test our system in a community-standardized evaluation, and on a corpus of crawled news articles from eight news sites to demonstrate the applicability to the web. Both experiments show significant improvement over baseline systems.

We discuss our method of soft pattern generalization and matching in the next section. Section 3 describes the architecture of our definition generation system, including details of our application of PRF to automatically label the training data for soft pattern generalization. We describe our experiments with the system framework in Section 4 and complete the paper with a discussion of related work.

2. SOFT MATCHING PATTERNS

At the heart of many definition generation systems is a process of identifying and selecting definition sentences. Many of these systems base their sentence selection either wholly or partially on pattern matching. In previous work, hand-crafted [5, 9] or machine learned [2] rules played a crucial role. Definition patterns also exist in news articles. This is supported by the observation that journalists often give explanations to those terms or people unfamiliar to the public and that they write such introductions in a regular manner. For example, appositives are a common pattern used to introduce a term or a person in news (*e.g.*, "Gunter Bloebel, *a cellular and molecular biologist*, …"). To make their writing more appealing to the public, news writers often exhibit great variations in wording and structuring of such definitions. Traditional hard matching rules are too rigid to accommodate such diversified patterns in definition sentences.

In this study, we augment the soft matching method advocated by Nahm and Mooney [11] and apply it to the problem of extracting definition patterns. They represented patterns by simple lexical tokens and employed cosine similarity to match patterns, similar to the technique employed by Agichtein and Gravano [1]. We augment their approach by: a) combining lexical tokens alongside part-of-speech classes and punctuations; and b) adopting a probabilistic framework that combines slot content and sequential fidelity in computing the degree of pattern match.

2.1 Generalizing Pattern Instances

Given a group of potential definition sentences, our system learns local contextual patterns surrounding the given search term. We do not handle long-distance dependencies, as our observations show that definition sentences are identified mainly by adjacent words and punctuations.

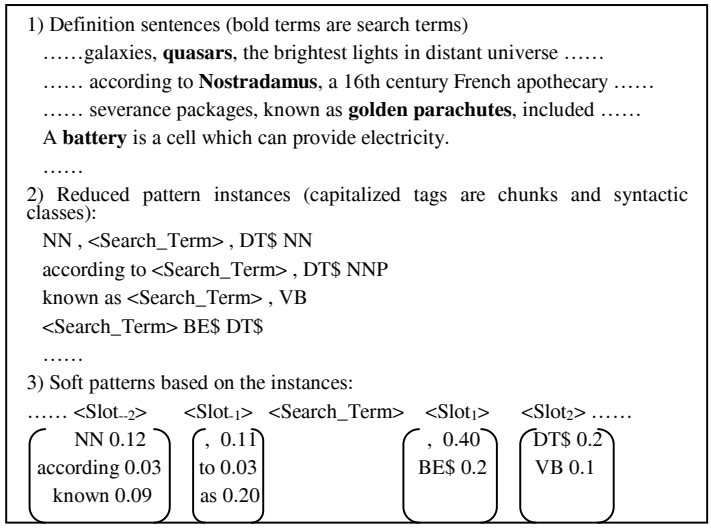

Figure 1. Illustration of generalizing soft definition patterns

The process of generalizing pattern instances is presented in Figure 1. The labeled definition sentences are first processed with part-of-speech (POS) tagging and chunking by a natural language tagger and chunker[1]. We then perform selective substitution of certain lexical items by their syntactic classes in order to generate representative patterns. The substitution attempts to replace words that are specifically related to the search term with more general tags so that the patterns can be applied to other sentences. The substitution rules that we use and some examples are listed in Table 1.

Table 1. Substitution heuristics.

Token	Substitution	Examples (from the example sentence in this section)
Any part of the search term	<SCH_TERM>	"Iqra" → <SCH_TERM>
Centroid Words: (Topical words related to the search term, detailed in section 3.1)	Corresponding syntactic classes	"channel" → NN
Noun phrases by chunking	NP	"Arab Radio and Television company" → NP
Adjectival and adverbial modifiers	*To be deleted*	
is, am, are, was, were	BE$	is → BE$
a, an, the	DT$	"the" → DT$
(all numeric values)	CD$	
All other words and punctuations	*no substitution*	"Owned", "by", "of", *etc.* are unchanged.

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

In Table 1, *centroid words* are those words that are highly correlated to the search term, as judged by mutual information. We explain this in depth in Section 3.1. The lexical forms of these words are too specific to the search term to help form general definition patterns and hence are replaced by their part-of-speech. Likewise, we perform the same substitution to noun phrases identified by chunking as different scenarios usually do not share the same noun phrase instances. Finally, we combine the adjacent syntactic tags of the same type into one. All other general words and punctuations are left unchanged.

Our algorithm is designed specifically to capture obscure patterns. To demonstrate this, we give an example of a definition pattern that is not likely to be covered by previous work. The example does not describe a direct definition but indicates some important properties of the search term, which should be included in its extended definition. Given a definition sentence for "Iqra":

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

the sentence is transformed into a token sequence comprising syntactic tags, words and punctuations as follows:

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

In order to generate general patterns, we need to consider the "context" around the <SCH_TERM>. The context is modeled as a window centered on <SCH_TERM> according to the pre-defined size *w*, *i.e.* the number of slots (or tokens) on both sides of <SCH_TERM>. Thus we get fragments with size *2w+1* including the search term. We refer to such fragments as *pattern instances* on which the generic soft patterns are generated. For example, the pattern instance from the above sentence is (*w=3*):

DT$ NN <SCH_TERM> BE$ owned by

Accumulating all the pattern instances extracted from the definition sentences and aligning them according to the positions of <SCH_TERM>, we obtain a virtual vector representing the soft definition patterns. The pattern vector *Pa* is denoted as:

$<Slot_{-w}, \ldots\ldots, Slot_{-2}, Slot_{-1}, SCH_TERM, Slot_1, Slot_2, \ldots\ldots Slot_w : Pa>$

where $Slot_i$ contains a vector of tokens with their probabilities of occurrence:

$<(token_{i1}, weight_{i1}), (token_{i2}, weight_{i2}) \ldots\ldots (token_{im}, weight_{im}) : Slot_i>$

Here $token_{ij}$ denotes any word, punctuation or syntactic tag contained in $Slot_i$; and $weight_{ij}$ gives the importance of the j^{th} token to the i^{th} slot. $weight_{ij}$ can be expressed as the conditional probability of the token occurring in that slot. Thus it can be approximated by:

$$\Pr(token_{ij} \mid Slot_i) = \frac{f(token_{ij})}{\sum_{s=1}^{m} f(token_{is})} \quad (1)$$

where $f(token_{is})$ stands for the number of occurrences of $token_{is}$ within $Slot_i$. As syntactic classes occur more frequently than lexical tokens, we discount the occurrences of syntactic classes and punctuations by a factor accounting for the proportion of words to syntactic tags. This discounting factor is used to achieve a good balance in the distribution and is empirically set to 0.1.

2.2 Soft Pattern Matching

What results from the generalization process is a virtual vector *Pa* with a set of associated probabilities for slot fillers at each slot. The soft pattern vector *Pa* is then used to calculate the degree to which a test sentence matches the sentences used to construct the soft patterns. The test sentences are first preprocessed with the identical procedures of POS tagging and chunking, as well as substitution as we did to the labeled definition sentences. Using the same window size *w*, the token fragment *S* surrounding the <SCH_TERM> is retrieved:

$<token_{-w}, \ldots\ldots, token_{-2}, token_{-1}, SCH_TERM, token_1, token_2, \ldots\ldots token_w : S>$

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, while the second part examines sequence fidelity. In the first part, we compute Pa_weight_{Slots} by assuming that all slots are independent to each other. We use Naïve Bayes to calculate the matching score:

$$Pa_weight_{Slots} = \Pr(S \mid Pa) = \prod_{i=-w}^{w} \Pr(token_i \mid Slot_i) \quad (2)$$

Specifically, we combine all the weights calculated in Equation (1) to derive the similarity for independent slots. This equation is very flexible in matching the soft patterns because it considers only individual slots. Even if some slots are missing, it still can give a similarity measure to the definition patterns.

The second part of the matching metric considers the *sequence* of tokens, to filter out unlikely token sequences to increase precision. We adopt a bigram model to formulate this sequence measure. Specifically, given a token sequence *T*, we calculate the conditional probability of $Pr(T|Pa)$ which models how likely the sequence occurs according to the underlying soft patterns. We calculate the sequence probability for the left and the right sequences starting from <SCH_TERM>. The probability of the right sequence is calculated as follows:

$$\begin{aligned} \Pr(right_seq \mid Pa) &= \Pr(token_1, token_2 \cdots token_w \mid Pa) \\ &= P(token_1)P(token_2 \mid token_1)\cdots P(token_w \mid token_{w-1}) \end{aligned} \quad (3)$$

where $P(token_i|token_{i-1})$ is estimated by counting the occurrences of the bigram $<token_{i-1}\ token_i>$ and the unigram $token_{i-1}$ as:

$$P(token_i \mid token_{i-1}) = \frac{f(<token_{i-1}\ token_i>)}{f(token_{i-1})} \quad (4)$$

The process for calculating the probability of the left sequence is formally identical. In addition, $P(token_{-1})$ and $P(token_1)$ can be estimated based on the proportion of occurrences of the token in

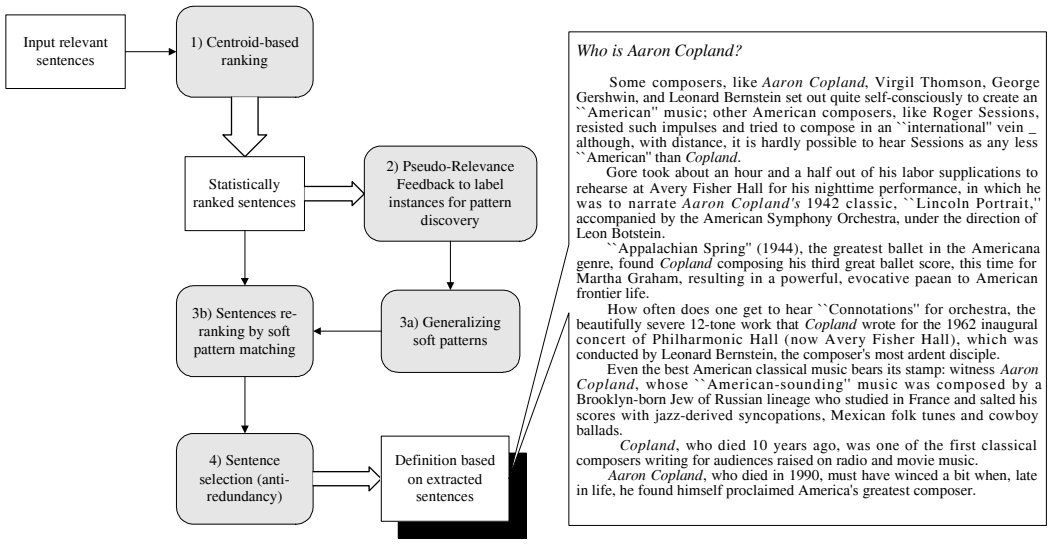

Figure 2. The architecture of our SP+PRF definition generation system and a sample sentence-extracted definition.

the immediately left and right slots to <SCH_TERM>. The sequence weight of the token vector for the sentence, denoted by Pa_weight_{Seq}, consists of the weights of its left sequence and right sequence which are calculated by Equation (3):

$$Pa_weight_{Seq} = (1-\alpha)\cdot \Pr(left_seq \mid Pa) + \alpha \cdot \Pr(right_seq \mid Pa) \quad (5)$$

Based on our observations of definition, the right context of the search term is more important in indicating a definition sentence, thus we set α to 0.7.

The bigram model may encounter the common problem of sparse data. But it is not a serious issue in our case because: (a) we use it just as a precision device to reduce the weight of those less possible sequences; and (b) we use large training set that we construct automatically using PRF, to be discussed in Section 3.2.

Finally, the aforementioned two similarity weights determine the overall pattern weight of the given sentence:

$$Pattern_weight = \frac{Pa_weight_{Slots} \times Pa_weight_{Seq}}{fragment_length} \quad (6)$$

where the length of the fragment *S* is used as the normalization factor.

3. DEFINITION GENERATION SYSTEM ARCHITECTURE

To demonstrate the effectiveness of soft matching patterns, we implemented a definition generation system which constructs definitions by extracting definition sentences from online news articles. The system's architecture and an example output definition for the question "Who is Aaron Copland?" are presented in Figure 2.

The input to the system is a set of relevant documents retrieved in response to a group of questions by an IR system. We then apply anaphora resolution to the documents and conduct passage retrieval to increase precision. The passage retrieval filters out all sentences that are not within a one-sentence window of a search term occurrence.

The steps in our system are outlined in Figure 3. All input sentences are first ranked statistically by a centroid-based statistical method. The system then takes the top *n* ranked sentences from the list and deems them as definition sentences (whether they are or not) in a pseudo relevance feedback loop. These automatically labeled sentences are fed into the pattern learning module where the soft patterns are generated. In the second round of ranking, the soft patterns and centroid weights jointly decide whether a sentence is labeled as definitional. Definition sentences from this final pool are selected to create the output definition using a diversity-based sentence selection algorithm. As the system's primary distinction from other definitional QA systems is its use of soft patterns and pseudo-relevance feedback, we denote it as SP+PRF in the remainder of the paper.

As soft pattern generalization and matching have been discussed in Section 2, we detail the remaining steps in this section.

Input: a set of questions and corresponding relevant sentences.

1. **First round of ranking (statistical ranking) –** Rank all input sentences statistically. In this work, we employ the centroid based method to accomplish the first round of ranking.
2. **Pseudo-relevance feedback –** Take all the top *n* ranked sentences (*n=10*) for each question from the statistical ranking as labeled definition sentences.

3a. **Soft pattern generalization –** Prepare the pattern instances and generalize the soft pattern vector from the pattern instances (see Section 2.1).

3b. **Second round of ranking (incorporating soft pattern matching) –** Re-rank the sentences combining the statistical centroid based weights and pattern matching weights (see Section 2.2).

4. **Sentence selection –** Produce the final definition according to the length requirement. Document summarization techniques are adopted to avoid introducing redundancy in constructing the definition.

Figure 3. Working process of the pipeline system

3.1 Step 1: Centroid Words Selection and Statistical Ranking

Given a set of input documents related to a person or term, our system first ranks its sentences in terms of their likelihood of containing a definition. To ensure recall, *i.e.* covering most aspects of the term or person, and to provide a basis for performing PRF, we first adopt a data-driven, centroid-based method to perform this ranking.

We identify a set of highly relevant topical words, which we term as *centroid words*. Similar to [13], we grouped the selected centroid words into a centroid vector, which is utilized to rank input sentences. Radev *et al*. [13] uses global TF×IDF weights within documents to select those words which are most representative across the entire documents. However, in our context, these centroid words should bear very specific information describing the search term. As such, we adopt a local centrality metric of words with respect to the search term based on their co-occurrences with the search term within sentences. The rationale is that the search terms tend to appear with their descriptive sentences within news articles. As a news article usually describes multiple terms and persons, descriptive sentences are likely to repeat the search term rather than using other forms of reference. Our sentences have also been processed by an anaphora resolution module. As such, co-occurrence based metric is able to capture the local importance of words to the search terms without losing recall.

The co-occurrences of words can be measured by using the metrics described in [8], which constructs topic signatures for document summarization. We employ mutual information as the measurement of co-occurrences for simplicity. All the words, after removing stop words, are stemmed before calculating their centrality. The equation for calculating the centrality $Centrality_{sch_term}(w)$ of a word w is as follows:

$$Centrality_{sch_term}(w) = \frac{\log(Co(w, sch_term) + 1)}{\log(sf(w) + 1) + \log(sf(sch_term) + 1)} \times idf(w) \quad (7)$$

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 inverted document frequency of w, $idf(w)$, as a measurement of its global importance[2].

Centrality scores for all words appearing in the input sentences are calculated and those words whose scores exceed the average plus a standard deviation form a set of centroid words. These centroid words form a centroid vector. Input sentences, which are also represented as vectors after tokenization and stemming, are compared against the centroid vector using cosine similarity. Sentences that rank highly are more likely to be definitional after this first pass.

3.2 Step 2: Unsupervised Labeling Using PRF

In order to perform soft pattern generalization, a set of labeled definition sentences should be provided as training instances, as is done in rule induction based on labeled data [15].

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

Step 1 automatically ranks sentences from the input documents, using words that are highly correlated with the search terms as indicators. To automatically decide whether a sentence is definitional, we could use a simple cutoff in which sentences that are ranked more highly are considered definitional. This is similar to the work by Sudo *et al.* [19] who proposed unsupervised learning method for pattern discovery by utilizing TF-IDF weight to select a set of relevant documents and sentences, and then built patterns from them.

Similarly, we use a pseudo-relevance feedback (PRF) strategy. In standard pseudo-relevance feedback (also known as blind feedback or local feedback) used in document retrieval, for each query, the top n ranked documents are deemed relevant and used to modify the query to retrieve a new set of documents [3]. We employ the same technique here: the system takes the top n ($n=10$) sentences from each question's ranking results and combine all the top ranked sentences for all questions as "blindly" labeled definition sentences. We then conduct the soft pattern generalization process on these sentences.

It is worth pointing out that we take all the top ranked sentences from a group of questions as a batch of labeled definition sentences which are fed into the pattern generalization module, instead of generalizing patterns from the results of one question. It makes the "blind" labeling process more reliable by constructing large training set to combat data sparseness.

One assumption here is that the top ranked list actually contains enough definition sentences that can be used to obtain good patterns. The other important assumption for group based PRF to work effectively is that the definition patterns derived from different questions are similar, which is reasonable for the domain of news. Thus, although some of the top ranked sentences for each search term are not definitional, the effects of such errors would be mitigated by performing PRF and pattern generalization over the entire group. Moreover, in journalistic text, descriptive sentences often contain essential information about the search term. Therefore some of the definition sentences will rank high by our centroid based method. This is supported by our experiments on TREC data. We observed that 33% of the top ten ranked sentences over a question set of 50 questions were actually definition sentences (165 of 500). While a 33% accuracy rate may seem low, it is still better than the baseline for performing PRF in [3]. Our experimental results show that the use of PRF significantly improves the quality of the resulting soft patterns.

3.3 Step 3b: Sentence Re-Ranking by Combining the Soft Pattern Matching Weight

The result of unsupervised pattern learning through PRF is a set of soft patterns as presented in Section 2 (Step 3a). We compute each input sentence's pattern matching weight by using Equation (6). The final score of a sentence incorporates both its centroid based weight and the soft pattern matching weight.

$$Def_Weight(stc) = (1-\delta) \cdot Centroid_weight(stc) + \delta \cdot Pattern_weight(stc) \quad (8)$$

where *Centroid_weight* denotes the statistical weight obtained by the centroid based method and *Pattern_weight* is the weight of soft pattern matching. δ represents a tunable parameter to favor either the centroid weight or the pattern weight. After an

initial study, we set it to 0.6, in order to give more weight to pattern rules because we believe that definition sentences should be sifted by patterns from the relevant sentences ranked by word statistics. Results shown later in this paper demonstrate that this combination of statistics and soft patterns is much more effective than using only the statistical method.

3.4 Step 4: Sentence Selection Module

In order to construct the final definition, one more step should be done to select the top ranked definition sentences according to the definition length requirement and to avoid introducing redundant sentences into the definition. We adopt a variation of Maximal Marginal Relevance (MMR) [4] to select non-redundant sentences from the top list of sentences ranked by Equation (8). The sentence selection algorithm is presented in Figure 4. Different from the approach taken in [4] that ranked all passages with MMR, our method examines only the top ranked sentences and stops when the length of the definition is satisfied. The reason of our adaptation is that the input sentences from news are very noisy and thus those sentences with lower weights are not reliable for generating definitions. Based on our study of the limited pilot questions provided in TREC-12 QA task, we return 7 sentences for terms ("What is X") and 10 sentences for people ("Who is X"). The heuristic is also employed in our submission to the TREC-12 definitional question answering task. Questions are classified using simple heuristics by looking at the head words [22].

(1) All sentences are ordered in descending order by weights.

(2) Add the first sentence to the definition pool.

(3) Examine the similarity of the next sentence *stc* in the remaining sentences to all sentences already in the definition pool. If *weight(stc) – average_similarity(stc, def_stc)* < *weight(next_stc)*, then skip sentence *stc*; otherwise add it to the definition pool. We use simple normalized word overlap to compute similarity.

(4) Repeat from Step 3 until the desired number of definition sentences is reached.

Figure 4. Sentence selection algorithm

4. EVALUATION

In this section, we report on two separate evaluations to show the effectiveness and adaptability of our SP+PRF system on the Web.

The purpose of our first experiment is to assess the effectiveness of our techniques in finding definitions from plain-text news articles on a publicly available standard corpus. We employ the TREC 2003 definitional question answering data set which includes a question set comprising 50 questions and answer judgments. We feel that the TREC QA corpus[3] is comparable to news articles found on the web, due to three reasons: a) all articles in the corpus are newswire articles; b) it is a corpus with a broad sample (~1 million articles), covering all kinds of topics; and c) definitions in web-based news articles tend to appear in the plain text of the article, and thus web markup tags and links do not help to distinguish them [9]. For these reasons, we believe that it is sufficient to use TREC data to justify the effectiveness of our method on the Web.

The purpose of the second experiment is to show the technique's adaptability to actual Web data and recent questions, as a proof-of-concept. We test the generality of our automatically-learned soft patterns. In this evaluation, we present the results obtained by our techniques on recently crawled online news ("Web corpus"). We collected 26 questions about people and events from the Lycos search engine, which were the most popular queries issued by users, during a day in September 2003. Most of the questions can be found in the Lycos 50[4] report. We list the 26 questions in Appendix 2. The questions were submitted to Google[5] to retrieve news articles within eight news sites, including BBC, CNN and USAToday. We set the limit for the number of pages downloaded from each site to 200. The text body of the news pages, embedded between the HTML tags "<P>" and "</P>", is extracted and preprocessed in the same fashion as was done to the TREC articles. We applied the learned soft patterns derived from the results of the 26 questions as well as those patterns learned from the TREC corpus to re-rank the sentences from the downloaded news articles.

The definition generation system used in the evaluation is illustrated in Figure 2. As we are not aware of a publicly-available comparable system, we used the system we developed for the TREC-12's definitional question answering task [22]. As the system employed hand-crafted rules, we denote it as HCR. The rules (listed in Appendix 1), partly derived from the previous work [9, 5], were carefully constructed for the TREC corpus. Specifically, HCR differs from the SP+PRF system in that: (i) it utilized hand crafted rules as in other existing work, instead of the soft pattern matching described in this work; and (ii) It used regular expressions to match the rules.

The HCR system achieved an F measure of 0.473 in TREC-12 evaluations. The system was ranked second according to TREC evaluations. Thus we have good reason to believe that HCR is representative of the state-of-the-art system in answering definition questions.

A careful reader may have noticed that machine learned rules by supervised learning are a good comparison target. We do not include them in our evaluations because the hand crafted rules in HCR were generalized by the developers manually after a long time and hence they can be good approximations of machine learned rules based on large amount of training data.

4.1 Evaluation Metrics

In order to get comparable evaluation results, we adopt the same evaluation metrics as used in TREC definitional question answering task [20]. For each question from the TREC corpus, there is a list of essential nuggets and acceptable nuggets for answering this question, provided by TREC. Using the given answer nuggets as a gold standard, an individual definition is scored using nugget recall (NR) and an approximation to nugget precision (NP) based on length. These scores are combined

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

[4] http://50.lycos.com.

[5] http://www.google.com.

using the F measure with recall being five times as important as precision. We list these metrics in Appendix 3.

As we do not have such standard nugget list for questions from the Web corpus, a total of seven different assessors were asked to mark definition sentences from the sentences returned by both the baseline and our method. For each question, an average of two assessors marked the resulting sentences. The NR, NP and F measures are calculated on the nuggets reflected in any of the definition sentences they have marked.

4.2 Effectiveness of the Proposed Techniques on TREC Corpus

In this evaluation, we compare the performance of our SP+PRF system against the HCR system on the 50 TREC questions. To illustrate the significance of definition patterns, the baseline system uses only the centroid based method (as in Section 3.1) to rank sentences. In the SP+PRF system, 683 pattern instances are extracted from the 500 blindly labeled definition sentences. We vary the window size *w* from 1 to 5 in soft patterns extraction and matching to study the impact of the distance of contextual slots from the search term. The results of NR, NP and F measures are listed in Tables 2, 3 and 4 respectively. We represent our techniques as "SP+PRF" with different window size settings.

From Tables 2, 3 and 4, we see significant improvements obtained by both the HCR and SP+PRF systems over the baseline statistical method, with the maximum improvement of 11.52% and 27.20% respectively for F measure. It shows that both the hand-crafted hard-coded rules as well as the automatically learned soft pattern rules are effective in selecting definition sentences. This is in line with our assumption that news articles define a term or person using some textual patterns.

We also see that a window size of 2 performs the best. This shows that definition patterns tend to be restricted to the tokens adjacent to the search term. The performance of our method drops when the window size reaches 4 or greater. Although a larger window size takes more contextual information into account, we believe it also introduces more noise in the distant slots. As phrase chunking and word omission have been done in the soft pattern generation process, we believe that the resulting small windows capture sufficient context.

The unsupervised SP+PRF system also outperforms the labor intensive HCR system. Over a man-month of time was used to develop the hand-crafted rules through continuous cycle of system coding and performance analysis. Despite a slight drop in precision for some window size settings, the recall and F measure obtained using our techniques are better than those by HCR, with a maximum improvement of 16.83% for recall and 14.06% for F measure for the window size of 2. A paired t-test gives the *p* values for the improvements in recall and F measure as 0.069 and 0.108, respectively.

We attribute such improvement to the soft matching patterns which are more flexible than hard coded crafted rules and are thus more adaptable to diversified patterns reflected in online news. Additional benefit comes from the feasibility of applying PRF to automatically labeling definition sentences for pattern discovery.

Table 2. Comparison of NR across the systems (TREC)

	NR	% improvement (over baseline)	% improvement (over HCR)
Centroid (Baseline)	0.463		
HCR	0.514	11.05%	
SP+PRF (w = 1)	0.561	21.14%	9.09%
SP+PRF (w = 2)	**0.601**	**29.74%**	**16.83%**
SP+PRF (w = 3)	0.579	25.16%	12.71%
SP+PRF (w = 4)	0.551	19.05%	7.21%
SP+PRF (w = 5)	0.557	20.33%	8.36%

Table 3. Comparison of NP across the systems (TREC)

	NP	% improvement (over baseline)	% improvement (over HCR)
Centroid (Baseline)	0.169		
HCR	0.206	22.05%	
SP+PRF (w = 1)	0.206	21.78%	-0.23%
SP+PRF (w = 2)	**0.221**	**30.94%**	**7.28%**
SP+PRF (w = 3)	0.217	28.24%	5.07%
SP+PRF (w = 4)	0.204	20.82%	-1.01%
SP+PRF (w = 5)	0.204	20.45%	-1.31%

Table 4. Comparison of F measure across the systems (TREC)

	F measure	% improvement (over baseline)	% improvement (over HCR)
Centroid (Baseline)	0.423		
HCR	0.472	11.52%	
SP+PRF (w = 1)	0.507	19.65%	7.29%
SP+PRF (w = 2)	**0.539**	**27.20%**	**14.06%**
SP+PRF (w = 3)	0.531	25.37%	12.42%
SP+PRF (w = 4)	0.495	16.97%	4.88%
SP+PRF (w = 5)	0.484	14.35%	2.54%

4.3 Evaluations on Online News in Web Corpus

In this evaluation, we present the results obtained by our techniques on the Web corpus. We first apply the centroid based method to ranking the sentences from the news pages, which is the baseline in the comparison. In the second configuration, we run the HCR system over the Web corpus. We applied our techniques to deriving 375 pattern instances through the PRF on the web corpus and used them to re-rank the sentences in constructing definitions. We denote this run as "SP+PRF (Lycos patterns)". In addition, we also utilize the 683 pattern instances derived from TREC corpus as soft patterns to re-rank the sentences. This run is represented as "SP+PRF (TREC patterns)".

Table 5. Comparison of NR, NP and F Measure for Web Corpus

	NR	% improvement (over baseline)	NP	% improvement (over baseline)	F Measure	% improvement (over baseline)
Centroid (baseline)	0.531		0.229		0.492	
HCR	0.598	12.67%	0.239	4.13%	0.555	12.82%
SP+PRF (Lycos patterns)	0.656	23.52%	0.277	20.82%	0.611	24.04%
SP+PRF (TREC patterns)	**0.682**	**28.35%**	**0.317**	**38.23%**	**0.642**	**30.33%**

Table 5 gives the results in terms of NR, NP and F measure for the baseline and our method with different sets of pattern instances. The window size of soft patterns is set to 2. The length of the definitions is the same as in the first evaluation.

From Table 5, we see significant improvements in results by our method over the baseline method. The improvements are of statistical significance. With the soft patterns learned from the results of the Lycos questions, the p values for the improvements in NR, NP and F measure are 0.0185, 0.0132 and 0.0161 respectively; while with the soft patterns from TREC corpus, the p values are even smaller, 0.0020, 0.0002 and 0.0013 respectively.

In addition, we re-affirm that the system employing soft patterns outperforms that with hand-crafted rules. Using soft patterns obtained from the TREC corpus gives the performance 15.52% higher (p=0.017) than that obtained by the HCR system.

It is noted that by using the soft patterns from the TREC corpus, the system performs better (5% higher in F measure) than that with the patterns learned from the Lycos questions' results. We construe that it is mainly due to the number of the pattern instances used in pattern generalization, which is 683 for the 50 TREC questions. This is twice as many as the 375 pattern instances derived from the 26 Lycos questions. The more pattern instances result in more generic definition patterns which are less affected by data sparseness.

The significant improvements by using the soft patterns derived from the TREC corpus also show that they are sufficiently portable to other sources of news articles. Pattern generality is largely achieved in our work by the proper substitution of search term specific words, as determined by centroid words. Appropriate window size is another important factor in avoiding introducing too much noise in learning the patterns.

5. RELATED WORK

Our work is most related to three streams of work: soft matching patterns, unsupervised rule induction and definitional question answering.

Soft matching has been utilized in information retrieval [16] where documents are matched by specified similarity measures. For textual tasks, such as classification [23], soft matching patterns that utilize word frequencies often perform better than hard-coded rules. Nahm and Mooney [11] proposed learning soft matching rules from text by combining rule-based and instance-based learning. Words in each slot are generalized by traditional rule induction techniques and testing instances are matched to the rules by their cosine similarities.

Information extraction usually relies on a set of specific rules [10]. Many supervised techniques have been suggested to learn extraction rules automatically, *e.g.* [18]. In order to relieve the labors in annotating corpus, some researchers started to address the problem of adaptive pattern discovery. Riloff [15] proposed to let users label entire sentences, rather than to tag the specific data to be extracted. The labeled sentences are used to obtain all word combinations in predefined syntactic relations. Similarly, Yangarber *et al.* [21] used a set of basic patterns as "seeds" and learn more scenario oriented extraction patterns automatically. Most relevant to our application of PRF, Sudo *et al.* [19] put forward an unsupervised learning for pattern discovery. They utilized TF×IDF to get a set of relevant documents and sentences and built patterns from them.

Answering definition questions is also addressed in previous work, especially in TREC. [12] used WordNet hyponyms to answer what-is questions. In the FALCON system, Harabagiu *et al.* [5] employed a simple yet widely used approach which defined several manually constructed definition patterns to extract proper phrases. Early TREC systems cannot deal with definition questions well due to the limitations of their simple techniques.

[17] and [2] proposed to combine data-driven statistical method and machine learned rules to generate definitions. The former is dedicated to producing biographical summaries for people, *i.e.* answering "who is" questions. They based the summary mainly on appositives and relative clauses. The latter tries to summarize definitional predicates of the given term to answer such questions. These two works cannot be applied to finding definitions from news because their rules of finding syntactic parts, like appositives and predicates, are too restrictive to be adapted to news articles.

More recently, the ubiquity of the Web has generated interest on finding definitions. Liu *et al.* [9] proposed mining topic specific definitions from the Web, but relied on a set of hand-crafted rules to find definition sentences. There is also work on extracting meaningful semantic components from online glossaries [6]. Our work differs from the above in that we seek to find the timely definitions for emerging terms and people, which are only present in breaking news and not in the more structured sources of information.

6. CONCLUSIONS

In this paper, we have proposed a set of techniques to construct definitions for newly emerging terms and people by extracting precise definition sentences in an unsupervised manner. Applying these techniques showed both an improvement in performance as well as cost saving in development time. Our work makes two significant contributions: First, we use soft matching patterns instead of hard-coded rules to select definition sentences. This technique is better suited to capture the diversity of definition patterns in news. Second, we introduce the application of pseudo-relevance feedback to perform

automatic labeling of training instances from ranked results. Our contribution here is to use PRF over a large set of input questions to counter noise and data sparseness. The automatically labeled definition sentences are utilized to generalize soft patterns. Experimental results show that our techniques outperform the state-of-the-art definitional question answering system without the need for an annotated corpus.

In future work, we plan to explore the application of soft patterns to information extraction and factoid question answering. Textual patterns [14] provide a simple yet effective way in finding answers for a question answering system. As existing factoid QA systems utilize surface textual rules, we believe soft patterns can improve the performance of such systems.

7. ACKNOWLEDGEMENT

The first author is grateful to Singapore Millennium Foundation Scholarship for supporting his PhD study (ref no. 2003-SMS-0230). The authors also thank Hwee Tou Ng for his valuable comments on this work.

8. REFERENCES

[1] E. Agichtein and L. Gravano, Snowball: Extracting relations from large plaintext collections. In Proceedings of the 5th ACM International Conference on Digital Libraries (DL'00), 2001, pp.85-94.

[2] S. Blair-Goldensohn, K.R. McKeown and A. H. Schlaikjer. *A Hybrid Approach for Answering Definitional Questions.* Technical Report CUCS-006-03. Columbia University, 2003.

[3] C. Buckley, G. Salton, J. Allan, and A. Singhal, *Automatic query expansion using SMART*, NIST Special Publication 500-225: The Third Text Retrieval conference (TREC 3), 1995, pp. 69-80.

[4] 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.

[5] 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.

[6] J.L.Klavans, S.Popper and R.Passonneau, *Tackling the Internet Glossary Glut: Automatic extraction and Evaluation of Genus Phrases,* In Proceedings of Semantic Web Workshop, SIGIR 2003, July 28 - Aug 1, 2003, Toronto, Canada.

[7] J.Lannon, 1991, *Technical Writing*, *Ch 5,* HarperCollins Publishers Inc., 1991.

[8] C.-Y. Lin and E.H. Hovy. *The Automated Acquisition of Topic Signatures for Text Summarization*. Proc. Of the COLING Conference. Strasbourg, France, 2000, pp.495-501.

[9] 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.

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

[11] U.Y. Nahm and R.J. Mooney, 2001, *Mining softmatching rules from textual data.* In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), pp. 979-984.

[12] J.M. Prager, D.R. Radev and K. Czuba, *Answering What-Is Questions by Virtual Annotation*, Proceedings of Human Language Technologies Conference, San Diego CA, pp. 26-30, March 2001.

[13] 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.

[14] 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.

[15] E. Riloff. *Automatically generating extraction patterns from untagged text*. In Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96), AAAI Press, 1996, pp. 1044-1049.

[16] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

[17] 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.

[18] S. Soderland, *Learning Information Extraction Rules for SemiStructured and Free Text*, Machine Learning: Special Issue on Natural Language Learning, 34, pp. 233-272.

[19] K. Sudo, S. Sekine, and R. Grishman. *Automatic Pattern Acquisition for Japanese Information Extraction*. Proc. HLT 2001, San Diego, CA, 2001.

[20] E.M.Voorhees, *Evaluating Answers to Definition Questions*, In Proceedings of HLT-NAACL 2003, pp. 109-111.

[21] R. Yangarber, R. Grishman, P. Tapanainen and S. Huttunen. *Automatic Acquisition of Domain Knowledge for Information Extraction*. Proc. 18th Int'l Conf. on Computational Linguistics (COLING 2000), Saarbrucken, Germany, July-August 2000, pp. 940-946.

[22] H. Yang, H. Cui, M-Y Kan, M. Maslennikov, L. Qiu, T-S Chua, *QUALIFIER in TREC-12 QA Main Task*, In Proceedings of the Twelfth Annual Text Retrieval Conference (TREC- 12), NIST, November, 2003.

[23] Y. Yang, *An evaluation of statistical approaches to text categorization*. Information Retrieval, Vol. 1, Number 1-2, 1999, pp. 69-90.

APPENDIX

Appendix 1. Hand crafted rules used in HCR.

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 characters.

Appendix 2. The 26 questions for the evaluation on Web corpus.

Question ID	Questions
1	Who is Brooke Burke?
2	Who is Clay Aiken?
3	Who is Jennifer Lopez?
4	What is Lord of the Rings?
5	Who is Pamela Anderson?
6	What is Hurricane Isabel?
7	What is Final Fantasy?
8	Who is Harry Potter?
9	Who is Carmen Electra?
10	What is Napster?
11	What is Xbox?
12	Who is Martha Stewart?
13	Who is Osama bin Laden?
14	What is Cloning?
15	What is NASA?
16	Who is Halle Berry?
17	What is Enron?
18	What is West Nile Virus?
19	What is SARS?
20	Who is Daniel Pearl?
21	Who is Nostradamus?
22	Who is James Bond?
23	Who is Arnold Schwarzenegger?
24	Who is Mohammed Saeed al-Sahaf?
25	Who is Uday Hussein?
26	What is stem cell?

Appendix 3. The evaluation metrics

NR = # essential nuggets returned in response / # essential nuggets

NP is defined using

allowance = 100 * (# essential + acceptable nuggets returned in response)

length = total # non-white-space characters in answer strings

NP = 1 if length < allowance

else 1-[(length – allowance) / length]

F = (26 * NP * NR) / ((25 * NP) + NR)