Learning Web Query Patterns for Imitating Wikipedia Articles

Shohei Tanaka[†]
tanaka@mi.ci.i.u-tokyo.ac.jp

Naokaki Okazaki[‡]
okazaki@is.s.u-tokyo.ac.jp

Mitsuru Ishizuka[†]
ishizuka@i.u-tokyo.ac.jp

[†]Graduate School of Information Science and Technology
University of Tokyo

[‡]Interfaculty Initiative in Information Studies
University of Tokyo

Abstract

This paper presents a novel method for acquiring a set of query patterns to retrieve documents containing important information about an entity. Given an existing Wikipedia category that contains the target entity, we extract and select a small set of query patterns by presuming that formulating search queries with these patterns optimizes the overall precision and coverage of the returned Web information. We model this optimization problem as a weighted maximum satisfiability (weighted Max-SAT) problem. The experimental results demonstrate that the proposed method outperforms other methods based on statistical measures such as frequency and point-wise mutual information (PMI), which are widely used in relation extraction.

1 Introduction

Wikipedia[1] is useful for obtaining comprehensive information of entities and concepts. However, even with 3.3 million English articles, Wikipedia does not necessarily include articles about an entity and concept of interest to a user. The ultimate goal of this study is to generate articles about an entity of a specified category from the Web by using Wikipedia articles in the same entity category as *exemplars*.

This study follows previous work of the other authors on query-biased/focused summarization (Tombros and Sanderson, 1998; Berger and Mittal, 2000) for modeling the target article generation process. In that model, when a user inputs an entity of interest, Web pages are retrieved that describe the entity by issuing queries to an information retrieval system. Using the retrieved pages as an information source, an article (summary) can be produced specialized for the target entity. From the application point of view, the article should include the concepts that best describe the target entity. In addition, articles concerning the entities of a category should cover concepts that are typical of the category. For example, an article about an actor is expected to mention his nationality, date of birth, movie credits, awards, etc.

A great number of researchers have addressed the problem of query-focused summarization (Carbonell and Goldstein, 1998; White et al., 2003; Dang, 2005; Daumé and Marcu, 2006; Varadarajan and Hristidis, 2006; Fuentes et al., 2007; Gupta et al., 2007; Wang et al., 2007; Kanungo et al., 2009). However, these studies assume that a document collection is provided for the summarization systems. In other words, collecting source documents that include important concepts for the target entity is not in the scope of these studies. For example, queries such as "(actor) was born in," "(actor) born on," "(actor) plays," and "(actor) won" may be more suitable than the simple query "(actor)" for obtaining concepts concerning the actor.

Source documents can be collected by a similar idea in relation extraction, which extracts entities having specific relations with the target entity (Hearst, 1992; Brin, 1999; Agichtein and Gravano, 2000; Turney, 2001; Pantel and Pennac-

[1] http://en.wikipedia.org/

Coling 2010: Poster Volume, pages 1229–1237,
Beijing, August 2010

chiotti, 2006; Blohm et al., 2007). These studies typically use statistical measures, such as frequency and point-wise mutual information (PMI), to assess the scores of the query patterns. However, these studies cannot eliminate the redundancy of concepts retrieved by a query set because they are designed to extract entities for each relation independently. For example, the query "(actor) born on" would not be necessary if the query "(actor) was born in" could gather documents referring to both the actor's nationality and date of birth.

In this paper, we propose a novel method for acquiring a set of high-quality query patterns that can gather source documents referring to important concepts about a specified entity. Given a category in which the entity is expected to be included, we use existing Wikipedia articles in this category to extract query patterns so that, when used together with the entity, they can retrieve important concepts related to the entity. We then select a small subset of query patterns that maximize the coverage and precision of the query result by modeling the query selection task as a weighted maximum satisfiability (weighted Max-SAT) problem.

2 Proposed method

First, let us define the terminology used in this paper. An *entity* is a topic for which we need to obtain an article (summary). Note that this definition is different from that used in other studies (e.g., named entity recognition). A *concept* is a noun phrase that has a specific relation to an entity. A *query pattern* is a lexical pattern that contains a slot filled by an entity. Used with an entity, a query pattern instantiates a query that collects related concepts. For example, "X was born in" is a query pattern in which X is a slot. When replacing X with an entity (e.g., "Dustin Hoffman"), the query pattern instantiates a query that may return the birthplace.

The goal of this study is, for a given entity category (e.g., *American actor*), to acquire a set of query patterns (*template*) for collecting related concepts from the Web. We learn the template by using Wikipedia articles of the category as supervision data. The method consists of three steps.

1. **Triplet extraction** identifies, for each Wikipedia article, entity mentions, concepts, and phrases that form a bridge between the entity mentions and concepts. In the context of learning query patterns from Wikipedia, we assume that a Wikipedia article is written for an entity. By identifying entity mentions and concepts in the article, we obtain bridging phrases between entity mentions and concepts as candidates for query patterns.

2. **Pattern assessment** verifies whether each candidate query pattern can actually retrieve concepts from the Web. This step issues queries of the form "(entity) (pattern)" to an information retrieval system, analyzes the retrieved Web pages, and examines whether each concept is found in the same sentence as the query expressions.

3. **Pattern selection** obtains a template by choosing a small subset of patterns such that the retrieved Web pages contain as many kinds of concepts as possible. We also eliminate query patterns that can retrieve descriptions other than concepts. We formalize this step as a weighted Max-SAT problem.

2.1 Triplet extraction

We first analyze Wikipedia articles to extract triplets of entities, query patterns, and concepts. Because a Wikipedia article usually describes a single entity, we identify the entity from the title of the article. We then search for occurrences of the entity in the body of the article. However, we might need to resolve coreference expressions because the entity might be described by a number of surface variations. For example, the Wikipedia article titled "Dustin Hoffman" might refer to the entity using "he" and "Hoffman" as well as "Dustin Hoffman"; the entity "Microsoft Corporation" might be described by "Microsoft" and "the company" in the article.

In general, coreference resolution is a non-trivial NLP task. Fortunately, Wikipedia articles are written for target entities. Therefore, we replace the occurrences of the following expressions in the body with the entity name:

Hoffman *was born in* [Los Angeles], [California], the second and youngest son of Lillian and Harry Hoffman, a [Russian]-born father who worked as a prop supervisor/set decorator at [Columbia Pictures] before becoming a furniture salesman. Hoffman *is from a* [Jewish] family, although he did not have a religious upbringing. He *graduated from* [Los Angeles High School] in 1955. He *enrolled at* [Santa Monica College] with the intention of studying medicine but left after a year to join the [Pasadena Playhouse].

Figure 1: A snippet of a Wikipedia article about "Dustin Hoffman."

1. Any token (split by spaces) that appears in the title of the article.

2. The phrase that appears the most frequently with the four anaphoric expressions "he," "she," "they," and "the *noun*."

The first rule deals with anaphoric expression caused by an ellipsis, e.g., "Dustin Hoffman" is referred to by "Dustin" and "Hoffman." The second rule resolves the coreference expressions caused by pronouns and definite noun phrases.

After detecting the entity mentions, we identify the concepts concerning the entity in the article. In this study, we employ WikiLink texts (anchor texts linked with other Wikipedia articles) that co-occur with the entity mentions in the same sentences. Finally, we identify a candidate of a query pattern as a phrase that satisfies the following conditions:

1. It consists of alphanumeric letters and hyphens only.

2. Its length is no longer than 6 tokens.

3. It appears between an entity mention and a concept in a sentence.

Figure 1 shows a snippet of the Wikipedia article about "Dustin Hoffman." The underlined expressions are identified as entity mentions; the text in square brackets represents a WikiLink text. Because all WikiLink texts appear in sentences with the entity mentions, we identify all expressions with square brackets as concepts. Italic texts are candidates of the query patterns.

Finally, we extract triplets of the form $\langle E_k, P_i, C_j \rangle$ from the Wikipedia article, where E_k ($k \in \{1,...,L\}$) denotes the entity, P_i ($i \in \{1,...,M\}$) denotes a query pattern, and C_j ($j \in \{1,...,N\}$) denotes a concept. For each concept C_j found in the Wikipedia article, we build a triplet by setting E_k as the entity of the article and P_j as the query pattern that precedes the concept C_j. Repeating this process for L Wikipedia articles in the same category, we obtain triplets with M query patterns and N concepts.

Table 1: Triplets extracted from Figure 1.

Entity	Query pattern	concept
Dustin Hoffman	was born in	Los Angeles
Dustin Hoffman	was born in	California
Dustin Hoffman	was born in	Russian
Dustin Hoffman	was born in	Columbia Pictures
Dustin Hoffman	is from a	Jewish
Dustin Hoffman	graduated from	Los Angeles High School
Dustin Hoffman	enrolled at	Santa Monica College
Dustin Hoffman	enrolled at	Pasadena Playhouse

Table 1 shows the eight triplets obtained from Figure 1. Here, it might not be clear whether the indefinite article *a* is necessary in the pattern *is from a*. Although we do not address this issue directly in this paper, we determine the popularity and usefulness of the pattern by analyzing Wikipedia articles in the same category. Similarly, some concepts (e.g., *Russian*) are not so important for the entity. It may be better to filter out the concept, but we expect that errors in concept identification are negligible when selecting the query patterns.

2.2 Pattern assessment

In this step, we verify whether each pattern P_i can actually retrieve concepts of the entities from the Web. More specifically, for every combination of an entity mention E_k and a pattern P_i, we issue a query "E_k P_i" (e.g., *"Dustin Hoffman graduated from"*) to Yahoo! Search BOSS[2]. We download the top 10 Web pages retrieved by each query and examine whether any of the concepts, C_j, appear in the same sentence as the query phrase. To describe the capability of the patterns for retrieving concepts, we introduce an $m \times n$ matrix called R,

$$R_{ij} = \begin{cases} 1 & \text{(pattern } P_i \text{ can retrieve concept } C_j\text{)} \\ 0 & \text{(otherwise)} \end{cases}.$$

[2]http://developer.yahoo.com/search/boss/

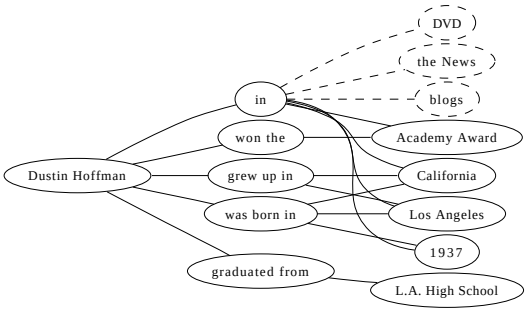

Figure 2: Patterns – collected concepts graph.

Figure 2 illustrates a bipartite graph between the patterns and concepts with R as the biadjacency matrix. Here, nodes with dotted lines are expressions other than concepts but are retrieved by the patterns. For example, the pattern "in" can retrieve four concepts *Academy Award*, *California*, *Los Angeles*, and *1937*, but it also retrieves non-concepts such as *DVD*, *the News*, and *blogs*. In other words, this pattern can retrieve sentences with a number of concepts, but it also gathers unnecessary sentences. Thus, we define the error rate of pattern P_i

$$\varepsilon_i = 1 - \frac{(\text{\# sentences with concepts})}{(\text{\# total sentences retrieved by } P_i)}.$$

2.3 Pattern selection

Based on the pattern assessment in Section 2.2, this step chooses a small set of query patterns as the template. Let $w_1, ..., w_m$ denote m Boolean (0–1 integer) variables, each of which (w_i) indicates whether the corresponding query pattern P_i is selected (1) or unselected (0). Choosing a subset of query patterns is equivalent to assigning Boolean values to the variables $w_1, ..., w_m$. The number of selected patterns is $\sum_{i=1}^{m} w_i$.

Given an assignment of variables $w_1, ..., w_m$ for the query patterns, we can examine whether the concept C_j is retrieved from the patterns by using the logical sum,

$$c_j = w_1 R_{1j} \vee w_2 R_{2j} \vee ... \vee w_m R_{mj} = \bigvee_{i=1}^{m} w_i R_{ij}.$$

Here, c_j is a Boolean (0–1) variable indicating that concept C_j is retrieved (1) or not retrieved (0) by the template. In Figure 2, if either the "in" or "was born in" pattern is selected, we can retrieve the concept "*1937*" from the Web search.

To choose a set of query patterns, we maximize the number of concept coverages $\sum_{j=1}^{n} c_j$ as well as minimize the number of patterns selected $\sum_{i=1}^{m} w_i$ and the total of the error rates of the selected patterns $\sum_{i=1}^{m} \varepsilon_i w_i$. This is achieved by solving the following problem.

Problem 1.

$$\begin{aligned} &\textit{Maximize} \sum_{j=1}^{n} c_j - \alpha \sum_{i=1}^{m} w_i - \beta \sum_{i=1}^{m} \varepsilon_i w_i, \\ &\textit{subject to: } c_1 = \bigvee_{i=1}^{m} w_i R_{i1} \\ &\qquad \cdots \\ &\qquad c_n = \bigvee_{i=1}^{m} w_i R_{in}, \\ &\qquad w_i \in \{0,1\}. \end{aligned}$$

Here, α and β are the parameters for controlling the preference of a smaller number of patterns (α) and the preference of accurate patterns (β).

To solve this problem, we rewrite it as a weighted maximize satisfiability (weighted Max-SAT) problem.

Problem 2.

$$\begin{aligned} &\textit{Maximize} \sum_{k=1}^{n+m} \lambda_k x_k \\ &\textit{Subject to: } x_1 = \bigvee_{i=1}^{m} w_i R_{i1} \quad (\lambda_1 = 1) \\ &\qquad \cdots \quad (\ldots) \\ &\qquad x_n = \bigvee_{i=1}^{m} w_i R_{in} \quad (\lambda_n = 1) \\ &\qquad x_{n+1} = \neg w_1 \quad (\lambda_{n+1} = \alpha + \beta\varepsilon_1) \\ &\qquad \cdots \quad (\ldots) \\ &\qquad x_{n+m} = \neg w_m \quad (\lambda_{n+m} = \alpha + \beta\varepsilon_m) \\ &\qquad w_i \in \{0,1\} \end{aligned}$$

Instead of subtracting the penalty terms from the objective value, we give bonus weights ($\alpha + \beta\varepsilon_i$) if the pattern P_i is not selected. This is achieved by introducing additional clauses x_{n+1}, ..., x_{n+m} that are satisfied by $\neg w_1$, ..., $\neg w_m$, respectively. Therefore, the optimization process tries to find a compromise between selecting patterns (clauses x_1, ..., x_n) and rejecting patterns (clauses x_{n+1}, ..., x_{n+m}). Although the complexity of the weighted Max-SAT problem is NP-hard, we use MiniMaxSAT (Heras et al., 2008) to solve the problem.

3 Evaluation

To verify the performance of the proposed method, we compare the precision, coverage, and F'-score of the information retrieval process by using the template obtained by the proposed method with that by three other baseline methods.

3.1 Experimental Settings

3.1.1 Data

We use articles of five categories in Wikipedia as the data for evaluation: *American actors*, *Genetic disorders*, *American tennis players*, *Software companies*, and *Operas*. Among these categories, the first two (*American actors*, *Genetic disorders*) have been commonly used as evaluation data in previous research on text summarization (Sauper and Barzilay, 2009). The other three (*American tennis players*, *Software companies*, *Operas*) are categories about three distinct topics (*sport*, *business*, *entertainment*).Table 2 shows information about these categories.

We divide the article set of a given category into six subsets. We use one subset as the *development set* for tuning the parameters α and β in the proposed method. The remaining five subsets are the *training set* and the *test set*, which are used for the 5-fold cross-validation. We create the template by using the training set and evaluate it with the test set. For evaluation of the baseline methods, we use only the *training set* and the *test set*.

3.1.2 Baselines

Random Selection

The baseline "*Random Selection*" randomly selects 10 query patterns from the candidate query patterns as the template for the category.

Frequency

In this baseline method, we sort the query patterns for each category in the order of frequency of occurrences in the category. We then select the top 10 frequent query patterns as the template for the category.

PMI-Web

The baseline *PMI-Web* chooses the query patterns that are the most "reliable." Following *KnowItNow* (Cafarella et al., 2005) and *Espresso* (Pantel and Pennacchiotti, 2006), the "reliability" of a pattern is defined by using the strength of the association of the pattern with the entities and concepts co-occurring with the pattern. In *KnowItNow* and *Espresso*, PMI (point-wise mutual information) is used to measure the strength of this association. PMI is estimated with the Web search hit counts as follows:

$$pmi(E_k, P_i, C_j) \approx \frac{\mathrm{hit}(E_k, P_i, C_j)}{\mathrm{hit}(E_k, P_i) \cdot \mathrm{hit}(C_j)},$$

where $\mathrm{hit}(E_k P_i)$, $\mathrm{hit}(C_j)$ are the Web search hit counts for the query "E_k, P_i," "C_j"(E_k, P_i, C_j is *entity, pattern, concept*), and $\mathrm{hit}(E_k, P_i, C_j)$ is the hit count for the query "E_k P_i" and "C_j." The reliability score of the query pattern is defined as the following formula:

$$Score(P_i) = \frac{1}{|S|} \sum_{(E_k, C_j) \in S} pmi(E_k, P_i, C_j),$$

where S is the set of pairs of *entity* E_k and *concept* C_j co-occurring with the *pattern* P_i in a sentence. The method *PMI-Web* chooses the top 10 patterns that have the highest reliability scores.

3.1.3 Experiments

We use each method to generate a template and retrieve information of the entities by using the

Table 2: The five categories used for evaluation.

Category	#Articles	#Patterns	#Concepts
American actors	1864	2951	10495
American tennis players	444	1039	2826
Software companies	1890	1992	5087
Genetic disorders	657	1087	2400
Operas	1425	2125	6365

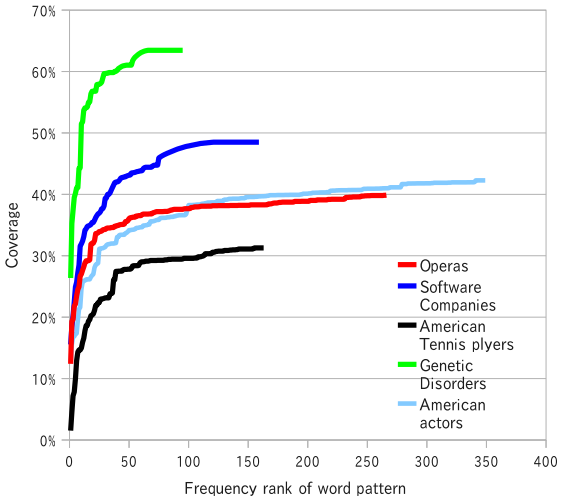

Figure 3: Relation between coverage and query pattern frequency.

query patterns in the template. We remove the query patterns occurring only once in each category from the candidate patterns because these patterns may be too entity-specific or noisy.

Figure 3 shows the coverage of the concept retrieval process when we use the top N frequently appearing patterns in the candidate pattern set. We observe that the coverage does not reach 100% even if we use all the query patterns. This is because some concepts cannot be retrieved by any query pattern. Moreover, we consider a Wikilink as a concept, even though some Wikilink texts do not actually represent a concept.

We use query patterns that occur no less than 3 times (for American actors, American tennis players, Software companies, and Operas) or twice (for Genetic disorders) in the corresponding Wikipedia articles so that the query patterns reach 95% of the upper bound of the coverage. This small subset comprises the final candidate patterns. For the candidate patterns, we use the proposed method (solving the weighted Max-SAT problem) and the three baselines described above to choose N query patterns.

The precision, coverage and quasi F-score (F'-score) of the information retrieval process by each template are defined as follows:

$$\text{precision} = \frac{\text{freq}(\text{E}_\text{k}, \text{P}_\text{i}, \text{C}_\text{j})}{\text{freq}(\text{E}_\text{k}, \text{P}_\text{i})}, \text{coverage} = \frac{\text{C}_\text{collected}}{\text{C}_\text{total}},$$

$$F' = \frac{2 \cdot \text{precision} \cdot \text{coverage}}{\text{precision} + \text{coverage}},$$

where $\text{freq}(E_k, P_i)$ is the frequency of the phrase "E_k P_i" in the retrieved documents, $\text{freq}(E_k, P_i, C_j)$ is the frequency of co-occurrence of the phrase "E_k P_i" and "C_j" in the sentences. C_total is the total number of distinct concepts in the data set, and $\text{C}_\text{collected}$ is the number of distinct concepts which can be collected by the template.

3.2 Result

Table 5 shows the average of the precision, coverage and F' scores of the five categories when we choose 10 query patterns (N=10). The proposed method obtains the highest score of all methods. Moreover, the proposed method outperforms the baselines not only for the average of all categories, but also for each category. This result indicates that the proposed method is able to choose query patterns that precisely and comprehensively collect the target concepts.

Table 5: Performance of the templates produced by the proposed method and the three baselines (N=10).

Method	Precision	Coverage	F' score
Random	16.56	11.40	13.19
Frequency	21.43	29.29	24.40
PMI-Web	22.55	22.08	21.42
Proposed	**27.34**	**30.77**	**27.95**

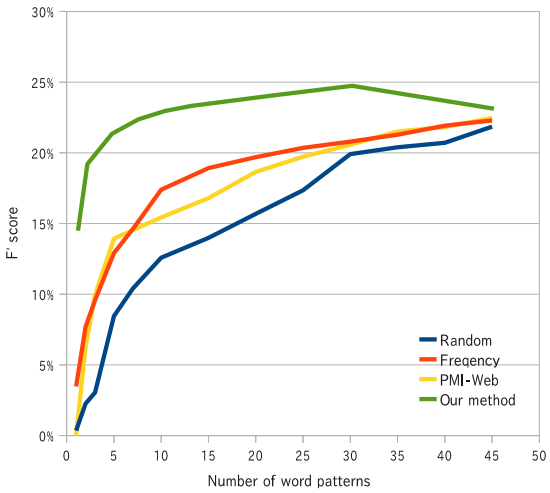

Figure 4: Number of query patterns (N) in template and F' score in *American tennis players*.

Table 3 shows some example templates produced by the proposed method. In this table, the number in the parentheses next to a pattern is the frequency rank of the pattern. We observe that the proposed method generates templates with two types of query patterns: *generic patterns* and *specific patterns*. Generic patterns such as "is a" and "is an" are patterns that can appear in every category. These patterns cover various kinds of concepts (high coverage), but may retrieve sentences that do not describe any concept (low precision). Specific patterns, such as "has a star on the" and "was nominated for a," can retrieve concepts that have specific relations with the entity. Therefore, queries with specific patterns retrieve a small number of concepts with high precision. The proposed method chooses query patterns in both of these types to achieve both high precision and high coverage. Therefore, it is able to retrieve

Table 3: Templates generated by the proposed method (N=10): (n) is the frequency rank.

Category	Template
American actors	"is an"(1), "was an"(2), "was a"(7), "graduated from"(9), "died of"(18), "has a star on the"(24), "was nominated for a"(28), "was married to"(47), "was born on"(56), "has appeared in"(92)
American tennis players	"defeated"(3), "beat"(5), "is a former"(12), "is an"(16), "graduated from"(17), "reached the"(18), "played"(24), "of"(30), "was"(38), "won"
Software companies	"is a"(1), "acquired"(3), "is"(9), "is headquartered in"(11), "was founded by"(15), "has offices in"(22), "was"(29), "include"(36), "introduced"(41), "is an international"(41)
Genetic disorders	"is a"(1), "is an"(2), "has an autosomal recessive pattern of"(3), "has an autosomal dominant pattern of"(9), "is named after"(11), "is a form of"(13), "is caused by"(16), "include"(18), "appears to be inherited in an"(41), "is considered an"(41)
Operas	"is an"(1), "is a"(2), "is an opera by"(17), "was"(18), "is a comic"(20), "premiered at the"(25), "was first performed at"(31), "is the second"(38), "opera"(46), "libretto by"(62)

Table 4: Templates generated by different methods for the *Opera* category (N= 10).

Method	Template	Pre.	Cov.	F'
Random	"was an," "of the complete operas of the," "was on," "was commissioned by," "by," "is," "popular," "for the," "New York," "the same name by"	15.79	8.12	10.73
Frequency	"is an," "is a," "the," "by," "of the," "in," "and," "of," "was a," "a"	16.83	27.12	20.77
PMI-Web	"was created by," "is a three act," "premiered at the," "was an," "is an," "is an opera composed by," "is a Hindi language," "premiered on" "was commissioned by," "was first performed at the,"	30.25	18.29	22.80
Proposed	"is an," "is a," "is an opera by," "was," "is a comic," "premiered at the," "was first performed at," "is the second," "opera," "libretto by"	31.18	27.28	29.10

various types of concepts. This implies that the method achieves high coverage even for concepts that cannot be retrieved by generic patterns.

The baseline *Frequency* obtains the second highest F'-score. It achieves high coverage but low precision. This is because this method chooses high-frequency patterns that can appear with every concept. Therefore, it is able to retrieve concepts with high coverage. However, these patterns do not retrieve specific information concerning a concept. Moreover, some high-frequency patterns, such as "the " and "by," lead to sentences that do not describe any concept.

Table 4 shows the patterns generated for the category *Opera* by each method. We can observe that the method *Frequency* chooses very generic patterns, such as "is a," "and," and "the," which are not specific to *Opera*.

In contrast, the method *PMI-Web* achieves high precision but low coverage. This is because this method chooses highly reliable patterns (e.g., "was commissioned by"), which are strongly associated with a specific kind of concept. However, these patterns cannot retrieve a broad range of concepts related to the target entity. This explains why the method cannot achieve high coverage.

Figure 4 shows the F' scores when we vary the number of selected query patterns (N) for the category *American tennis players*. We observe that the templates generated by the proposed method achieve the highest F'-score at every value of N. The maximum F'-score is 24.7, which is achieved when N is 30. Moreover, the proposed method requires only five query patterns to achieve the F'-score of 21.4. Therefore, the proposed method achieves a high F'-score by using only a small number of patterns. This implies that the method achieves high performance in a short query processing time.

4 Related Work

Many studies have addressed the problem of pattern extraction from Wikipedia (or other large corpora). Filatova et al. (2006) presented an approach for automatically extracting important word patterns from a large corpus. They analyzed the BBC corpus to extract word patterns containing verbs that are supposed to be important for a specific domain. Biadsy et al. (2008) described a system for producing biographies for a given target name. They used Wikipedia to learn the document structures of a biography. Ye et al. (2009)

explored a method for generating a series of summaries of various lengths by using information from Wikipedia.

Sauper and Barzilay (2009) proposed an approach for creating a summary of many chunks of text that are related to an entity and retrieved from the Web. They used Wikipedia not only for producing the template, but also for improving the summaries. Although the target of their work is very close to that of our study, the focus of each study is different. They address the method for selecting appropriate sentences for summarization, whereas we consider the method for selecting query patterns that can generate a comprehensive summary of an entity.

Various studies have addressed Web page summarization and query-focused summarization, from search result summarization (Kanungo et al., 2009) to query biased summarization (Wang et al., 2007). Furthermore, Fujii and Ishikawa (2004) presented a method to automatically compile encyclopedic knowledge from the Web.

Similar to relation extraction, the proposed method retrieves information concerning an entity by using query patterns. This is because query patterns for relation extraction are also appropriate in sentence extraction for multi-document summarization (Hachey, 2009). However, the relation extraction task primarily obtains query patterns that retrieve instances of a specific relation. This is different from the goal of this study, which is obtaining a set of patterns that are able to retrieve a large range of topics related to an entity.

5 Conclusion

We present a novel method to acquire a set of query patterns for retrieving documents that contain important information regarding an entity. Especially, we concentrate on the method for selecting query patterns that are able to comprehensively and precisely retrieve important concepts concerning an entity. The experimental results demonstrate that the proposed method outperforms methods based on statistical measures such as frequency and point-wise mutual information (PMI), which are widely used in relation extraction.

Currently, we use the text between an entity and a WikiLink as a candidate for a query pattern. In the future, we plan to use the text between two noun phrases as query patterns to increase the number of candidates for the pattern selection process. Moreover, we intend to build a text summarization application based on the proposed method to confirm that the selected pattern set is able to generate a comprehensive summary for an entity.

References

Agichtein, Eugene and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proc. of the fifth ACM conference on Digital libraries*, pages 85–94.

Berger, Adam and Vibhu O. Mittal. 2000. Query-relevant summarization using FAQs. In *Proc. of the 38th Annual Meeting on Association for Computational Linguistics*, pages 294–301.

Biadsy, Fadi, Julia Hirschberg, and Elena Filatova. 2008. An unsupervised approach to biography production using Wikipedia. In *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 807–815.

Blohm, Sebastian, Philipp Cimiano, and Egon Stemle. 2007. Harvesting relations from the Web: quantifiying the impact of filtering functions. In *Proc. of the 22nd national conference on Artificial intelligence*, pages 1316–1321.

Brin, Sergey. 1999. Extracting patterns and relations from the World Wide Web. *Selected papers from the International Workshop on The World Wide Web and Databases*, pages 172–183.

Cafarella, Michael J., Doug Downey, Stephen Soderland, and Oren Etzioni. 2005. KnowItNow: Fast, scalable information extraction from the Web. In *Proc. of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 563–570.

Carbonell, Jaime and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Dang, Hoa Trang. 2005. Overview of DUC 2005. In *Document Understanding Conference (DUC) 2005*.

Daumé, III, Hal and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proc. of the 21st*

International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pages 305–312.

Filatova, Elena, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proc. of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 207–214.

Fuentes, Maria, Enrique Alfonseca, and Horacio Rodríguez. 2007. Support vector machines for query-focused summarization trained and evaluated on pyramid data. In *Proc. of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 57–60.

Fujii, Atsushi and Tetsuya Ishikawa. 2004. Summarizing encyclopedic term descriptions on the Web. In *Proc. of the 20th international conference on Computational Linguistics*, pages 645–651.

Gupta, Surabhi, Ani Nenkova, and Dan Jurafsky. 2007. Measuring importance and query relevance in topic-focused multi-document summarization. In *Proc. of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 193–196.

Hachey, Ben. 2009. Multi-document summarisation using generic relation extraction. In *Proc. of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 420–429.

Hearst, Marti A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th conference on Computational linguistics*, pages 539–545.

Heras, Federico, Javier Larrosa, and Albert Oliveras. 2008. MiniMaxSat: An efficient weighted Max-SAT solver. *Journal of Artificial Intelligence Research*, 31:1–32.

Kanungo, Tapas, Nadia Ghamrawi, Ki Yuen Kim, and Lawrence Wai. 2009. Web search result summarization: Title selection algorithms and user satisfaction. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1581–1584.

Pantel, Patrick and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proc. of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120.

Sauper, Christina and Regina Barzilay. 2009. Automatically generating Wikipedia articles: a structure-aware approach. In *Proc. of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216.

Tombros, Anastasios and Mark Sanderson. 1998. Advantages of query biased summaries in information retrieval. In *Proc. of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–10.

Turney, Peter D. 2001. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *Proc. of the 12th European Conference on Machine Learning*, pages 491–502.

Varadarajan, Ramakrishna and Vagelis Hristidis. 2006. A system for query-specific document summarization. In *Proc. of the 15th ACM international conference on Information and knowledge management*, pages 622–631.

Wang, Changhu, Feng Jing, Lei Zhang, and Hong-Jiang Zhang. 2007. Learning query-biased Web page summarization. In *Proc. of the sixteenth ACM conference on information and knowledge management*, pages 555–562.

White, Ryen W., Joemon M. Jose, and Ian Ruthven. 2003. A task-oriented study on the influencing effects of query-biased summarisation in Web searching. *Information Processing and Management*, 39(5):707–733.

Ye, Shiren, Tat-Seng Chua, and Jie Lu. 2009. Summarizing definition from Wikipedia. In *Proc. of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*, pages 199–207.