Re-ranking answer candidates based on exhaustiveness of variety of answer viewpoints in non-factoid QA

Kotaro Sakamoto
Yokohama National University
sakamoto@forest.eis.ynu.ac.jp

Keiichi Nagao
Yokohama National University
nagao@forest.eis.ynu.ac.jp

Hayato Kobayashi
Yokohama National University
hayato-k@forest.eis.ynu.ac.jp

Hideyuki Shibuki
Yokohama National University
shib@forest.eis.ynu.ac.jp

Tatsunori Mori
Yokohama National University
mori@forest.eis.ynu.ac.jp

Noriko Kando
National Institute of Informatics
kando@nii.ac.jp

ABSTRACT

We propose a graph-based method for re-ranking answer candidates of non-factoid QA systems in terms of the coverage of the variety of answer viewpoints. The method uses an extended HITS algorithm and a graph model that has vertices of answer candidates and links from their text fragments to entire texts. Furthermore, the method merges fragments to represent a viewpoint appropriately. We conducted an experiment using the question set in the test collection of the NTCIR-6 QAC-4 task, and confirmed the effectiveness of the graph model and merging.

Keywords

re-ranking answer candidates, graph-based ranking, QA

1. INTRODUCTION

In general, the current non-factoid Web QA framework retrieves documents relative to a question, extracts texts appropriate to the answer from the documents, and ranks the texts in their order of appropriateness. Each extracted text may provide incomplete accounts of an answer because retrieved documents are written from various viewpoints such as definition, history, and characteristics, and because an extracted text cannot always include all viewpoints.

For example, to a question "What is the sport of skeleton?", the following three texts are adequate answers.

(a) Skeleton originated in St. Moritz, Switzerland, as a spinoff of the popular British sport called Cresta sledding. It was added to the Olympic program for the 2002 Winter Olympics; previously, it had been in the Olympic program only in 1928 and 1948.

(b) Skeleton racing involves plummeting head-first down a steep and treacherous ice track on a tiny sled. The rider experiences forces up to 5G and reaches speeds greater than 130 km/h (80 mph). It is considered the world's first sliding sport.

(c) Skeleton is a fast winter sliding sport, featured in the Winter Olympic Games, in which a person rides a small sled down a frozen track while lying face down (prone). The name of the sport originated from the bony appearance of the sled.

The texts have different viewpoints. For complete accounts, a user needs to read all of them.

Let us consider the order in which the texts are to be shown such that user gains the best understanding. Texts (a) and (b) go into details about a special viewpoint, such as history and characteristics, while text (c) is an overview including components from various viewpoints. Therefore, we consider that it is appropriate to show text (c) first and then show either or both the other texts according to the user's interest.

In this paper, we propose a method for re-ranking answer candidates based on the exhaustiveness of the variety of answer viewpoints. The method is based on graph theory, and answer candidates outputted by any non-factoid QA system as input can be accepted in this method. Although the method is independent of languages, we conduct an experiment using Japanese texts.

2. BASIC IDEA

When a set of answer candidates is given, finding an answer candidate including more components scattered over the answer candidates is similar to finding a more representative sentence in multi-document summarization. For sentence extraction in automatic summarization, graph-based methods such as TextRank[6] and LexRank[2] are known to be effective.

We use a graph-based algorithm for re-ranking answer candidates and focus on the HITS algorithm[4], which is also used in TextRank. The HITS algorithm calculates the hub and authority scores of vertices in a graph. A good hub is a vertex linked to many other vertices, and a good authority is a vertex linked from many different hubs. We regard an answer candidate including components mentioning many other answer candidates as a good hub, and we regard an

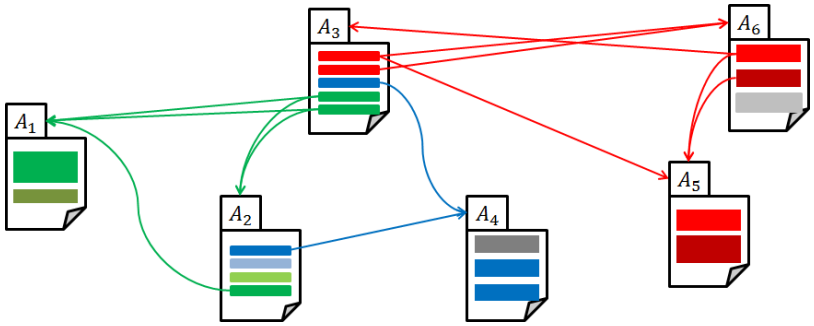

Figure 1: Graph model in the proposed method.

answer candidate mentioned by many different components of hubs as a good authority.

Figure 1 shows the graph model used in the proposed method. An answer candidate includes several text fragments, the sizes of which are suitable for representing a viewpoint, and a vertex is either an answer candidate or a fragment. Note that the hub and authority scores of an answer candidate are common to ones of fragments included in the answer candidate. A link can be drawn from a fragment to an answer candidate that must be different from one including the fragment, if the fragment represents a component of the answer candidate. Note that the start points of the link are limited to fragments and that the end points of the link are limited to answer candidates. In Figure 1, the vertex corresponding to answer candidate $A3$, which seems to mention various components of the other answer candidates, will have the greatest hub score because the vertex has the greatest number of links that are drawn to the other vertices.

The following three problems exist for automatically constructing the graph structure. The first problem is to divide answer candidates into fragments suited for representing a viewpoint. To solve this, answer candidates are temporarily divided into minimal units that can represent a viewpoint, and then adjacent fragments representing the same viewpoint are merged. The second problem is to judge whether fragments represent the same viewpoint. To solve this, we use the link structure of fragments. If fragments have the same or similar link structure, we regard the fragments as representing the same viewpoint. The third problem is to link fragments and answer candidates. Although TextRank and LexRank use links based on the similarity between sentences, the similarity does not work well between fragments and answer candidates, because their sizes are too different. Therefore, we use a measure based on textual inclusion instead of similarity.

3. ALGORITHM

3.1 Outline

Figure 2 shows the outline of the proposed method. The input is a set of answer candidates outputted by a non-factoid QA system. The method consists of the following four stages. The first stage involves fragmenting answer candidates into simple verb phrases, which are regarded as minimal units representing a viewpoint in Japanese. The second stage involves linking fragments to other answer candidates based on textual inclusion. The third stage involves merging adjacent fragments, the link structures of which are the same or similar. The fourth stage involves scoring answer candidates based on the HITS algorithm. The output is answer candidates re-ranked in the decreasing order of hub score.

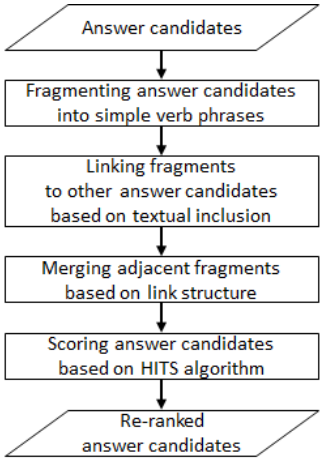

Figure 2: Outline of the proposed method.

3.2 Fragmenting answer candidates

We regard a simple verb phrase as a minimal unit that can represent a viewpoint in Japanese. To extract a simple verb phrases, we used the Japanese morphological analyzer Mecab[1] and Japanese dependency parser Cabocha[2][5].

3.3 Linking fragments to answer candidates

Because sizes between fragments and answer candidates are too different to estimate the association between them using similarity scores such as the cosine or Jaccard, we use a score with the asymmetric function of lexical overlap between them as a measure of the association. The lexical overlap score $sc_{lo}(F_{ij}, A_k)$ between a fragment F_{ij}, which is included in an answer candidate A_i, and an answer candidate A_k is calculated using the following equation:

$$sc_{lo}(F_{ij}, A_k) = \frac{|word(F_{ij}) \cap word(A_k)|}{|word(F_{ij})|} \quad (1)$$

[1]http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html
[2]http://chasen.naist.jp/chaki/t/2005-08-29/doc/CaboCha%20Yet%20Another%20Japanese%20Dependency%

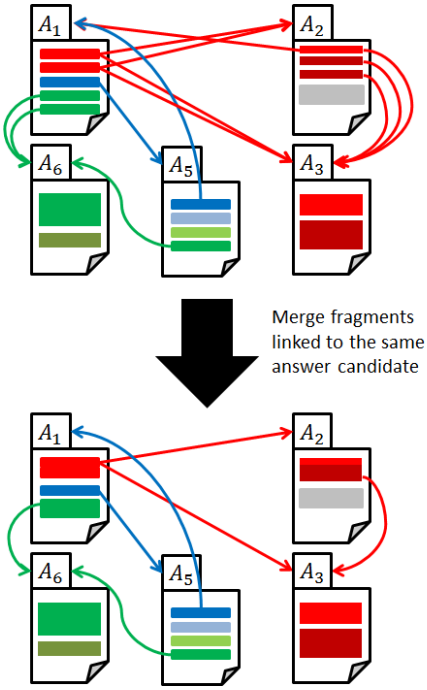

Figure 3: Necessity of merging fragments linked to the same answer candidate.

where $word(T)$ is a set of words in a text T. The score is used as a baseline for textual entailment[8]. If the score is greater than a threshold α, a link is drawn from the fragment to the answer candidate. We set α to 0.6 based on a preliminary experiment.

3.4 Merging fragments

Because the fragmentation described in 3.2 breaks answer candidates down into minimal units, different fragments in an answer candidate may represent the same viewpoint, i.e., they may link to the same answer candidate. A hub score of an answer candidate is calculated by summing authority scores indicated by fragments in the answer candidate. If different fragments in an answer candidate link to the same answer candidate as shown in the top of Figure 3, the authority score is multiplied with the link number for calculating the hub score. Because the multiplication has the effect of summing different authority scores, the HITS algorithm will not work well in terms of the coverage of various viewpoints. Therefore, fragments representing the same viewpoint should be merged as shown in the bottom of Figure 3.

Two adjacent fragments in an answer candidate are merged if their link structure matches either of the two patterns, as shown in Figure 4. In the first pattern, two fragments have the same set of answer candidates linked from the fragments. In the second pattern, a set of answer candidates linked from a fragment is a subset of answer candidates linked from the other fragment.

3.5 Scoring answer candidates

By using the graph structure after the merging described in 3.4, the HITS algorithm calculates the hub and authority

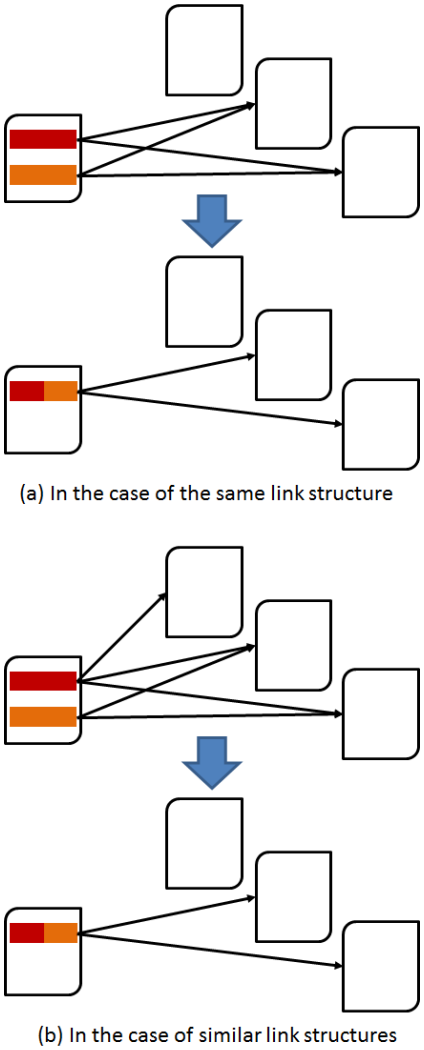

Figure 4: Patterns of link structures when fragments are merged.

scores of all answer candidates. Because the graph structure is different from the one used in TextRank, the hub score $sc_{hub}(A_i)$ and authority score $sc_{auth}(A_i)$ of an answer candidate A_i are, respectively, calculated using the following equations:

$$sc_{hub}^{t+1}(A_i) = \sum_{F_{ij} \in frag_L(A_i)} \frac{maxsc_{auth}^{t}(F_{ij})}{1 + \log(|frag_T(A_i)|)} \quad (2)$$

$$maxsc_{auth}(F_{ij}) = \max_{A_k \in ans_O(F_{ij})} sc_{auth}(A_k) \quad (3)$$

$$sc_{auth}^{t+1}(A_i) = \sum_{A_k \in ans_I(A_i)} \frac{sc_{hub}^{t}(A_k)}{1 + \log(|frag_L(A_i)|)} \quad (4)$$

where $flag_L(A_i)$ is a set of fragments linked to one or more answer candidates in A_i, $flag_T(A_i)$ is a set of all fragments in A_i, $ans_O(F_{ij})$ is a set of answer candidates linked from F_{ij}, and $ans_I(A_i)$ is a set of answer candidates with a fragment linked to A_i.

In equation (2), the division by the number of fragments is included to prevent the inappropriate increase of hub scores of long answer candidates, which tend to include many frag-

Table 1: MRR scores in the top 10 results

Method	MRR
TextRank	0.168
Proposed w/o merging	0.283
Proposed	0.575

ments. In equation (3), the selection of the maximum value is included to prevent the inappropriate increase of authority scores from fragments linked to many answer candidates. Because answer candidates representing fewer viewpoints seem to go into details about the viewpoints, we consider that texts associated with such answer candidates are also more representative of the viewpoints. This idea is introduced to equation (4) as the penalty of division by the number of fragments.

4. EXPERIMENT

4.1 Experimental Setup

We used the following two methods for comparison with the proposed method. The first method is the original TextRank based on the HITS algorithm, in which links are drawn between two answer candidates without fragmenting and merging. The second method is the proposed method without merging. By comparing the methods, we examine how the difference among the graph structures influences re-ranking.

We selected questions and answer candidates used in the experiment as follows. By using the Web QA system MinerVA[7], we collected the top 100 answer candidates per first 30 questions in the test collection of the NTCIR-6 QAC-4 task[3]. We checked whether there are answer candidates including components from several viewpoints and used questions involving such answer candidates. The number of questions was 11.

We evaluated the top 10 answer candidates re-ranked using both methods. If an answer candidate included several components from different viewpoints, it was judged as well re-ranked in terms of the coverage of various viewpoints. Otherwise, it was considered a poor re-ranking. We used the mean reciprocal rank (MRR) score[1] as the evaluation measure.

4.2 Results

Table 1 lists the MRR scores in the top 10 results for both methods. The table indicates that the graph model shown in Figure 1 was effective, and that the merging described in 3.4 could greatly improve the MRR score.

The errors are attributable to errors in morphological and syntactic analysis, linking failures due to mismatched expressions, merging failures bue to incorrect link structures, and scoring failures due to incorrect answer candidates. Analizing the errors is a task for the future.

5. CONCLUSION

We proposed a graph-based method for re-ranking answer candidates of non-factoid QA systems in terms of the coverage of the variety of answer viewpoints. The method uses an extended HITS algorithm and a graph model that has vertices of answer candidates and links from their text fragments to entire texts. Furthermore, the method merges fragments to represent a viewpoint appropriately. We conducted an experiment using a question set in the test collection of the NTCIR-6 QAC-4 task, and confirmed the effectiveness of the graph model and merging.

In future work, we will apply the method to generate a text covering answers from a particular viewpoint to a question. For example, in the NTCIR-11 QA-Lab task[9], a challenge to make QA systems answer essay questions of "world history" in real-world university entrance exams was conducted. Because the answer texts of the essay questions seem to consist of contents scattered in knowledge sources such as textbooks and Wikipedia, we consider that the method is suited for the task.

6. REFERENCES

[1] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM.

[2] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, Dec. 2004.

[3] J. Fukumoto, T. Kato, F. Masui, and T. Mori. An overview of the 4th question answering challenge (qac-4) at ntcir workshop 6. In *Proceedings of NTCIR-6*, 2007.

[4] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The web as a graph: Measurements, models, and methods. In *Proceedings of the 5th Annual International Conference on Computing and Combinatorics*, COCOON'99, pages 1–17, Berlin, Heidelberg, 1999. Springer-Verlag.

[5] T. Kudo and Y. Matsumoto. Fast methods for kernel-based text analysis. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 24–31, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[6] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In D. Lin and D. Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[7] T. Mori, M. Sato, and M. Ishioroshi. Answering any class of japanese non-factoid question by using the web and example q&a pairs from a social q&a website. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '08, pages 59–65, Washington, DC, USA, 2008. IEEE Computer Society.

[8] V. Rus, P. M. McCarthy, D. S. McNamara, and A. C. Graesser. A study on texual entailment. In *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI'05, pages 326–333, Washington, DC, USA, 2005. IEEE Computer Society.

[9] H. Shibuki, K. Sakamoto, Y. Kano, T. Mitamura, M. Ishioroshi, K. Y. Itakura, D. Wang, T. Mori, and N. Kando. Overview of the ntcir-11 qa-lab task. In *Proceedings of the 11th NTCIR Conference*, 2014.