TTOKU Summarization Based Systems at NTCIR-10 1CLICK-2 task

Hajime Morita
Tokyo Institute of Technology
morita@lr.pi.titech.ac.jp

Ryohei Sasano
Tokyo Institute of Technology
sasano@pi.titech.ac.jp

Hiroya Takamura
Tokyo Institute of Technology
takamura@pi.titech.ac.jp

Manabu Okumura
Tokyo Institute of Technology
oku@pi.titech.ac.jp

ABSTRACT

We describe our query-oriented summarization system implemented for the NTCIR-10 1CLICK-2 task. Our system is purely based on a summarization method regarding the task as a summarization process. The system calculates relevant scores of terms for a given query, then extracts relevant part of sentences from input sources. For the calculation of relevant scores for a query, we employed a Query SnowBall based method. In addition, the summarization is conducted as subtree extraction from dependency trees corresponding to the sentences in the input sources.

Keywords

Multi-document summarization, Submodular maximization, Greedy algorithm

Team Name: TTOKU
Language: Japanese
External Resources Used: none

1. INTRODUCTION

Automatic summarization has been studied for decades. Although newspaper articles have been chosen as the target text of summarization studies, Web texts become increasingly important as the target text, because Web texts are increasing at an explosive pace. In this paper, we describe a method that extracts the compressed and query relevant sentences. The method simultaneously conducts sentence extraction and sentence compression for a given query. We regard the NTCIR-10 1CLICK-2 task as a query-oriented summarization task, and generate a summary from the given input source and query. The query-oriented summarization is of growing importance in line with the development of information retrieval [12]. In the mobile situations, users often have only a small monitor and can hardly read long documents. The development of mobile technology increases the demands for short and quick response on the query-oriented summarization. Our joint model can generate a short and quick response by the sentence compression and fast greedy algorithm.

We generate summaries by solving a submodular maximization problem. *Submodular maximization* problems have long been studied, and recently, applied to the field of text summarization [7, 8]. Formalizing summarization as a submodular maximization problem has an important benefit that the problem can be solved by a greedy algorithm with performance guarantee. Therefore, the problem can be solved efficiently and we can easily estimate computational cost for the problem. The nature of submodular maximization is suitable for real-time applications.

The joint model of sentence extraction and compression can be formalized as an optimization problem with linear constraints. The direct formalization of joint model needs to treat numerous constraints, and thus the formalization is difficult to optimize as submodular maximization problem. We can handle the joint model as a submodular maximization with use of the formalization proposed in [10]. The joint model of sentence extraction and compression has a great benefit that it has a larger degree of freedom in changing the sentence length and thus controlling the redundancy. Since, in the mobile settings, it is vital to shorten the summary length, the sentence compression helps to pack more content in a short text. The nature of submodular maximization is also helpful for handling joint models efficiently, and improving the freedom of designing objective functions.

In this paper, we describe our system and the evaluation results on the NTCIR-10 1CLICK-2 task. Our system employs a variation of Query Snowball (QSB) [11] for query relevance scoring, and a subtree extractive approach for generating a non-redundant and short summary.

2. NTCIR-10 1CLICK-2 TASK

In main tasks of NTCIR-10 1CLICK-2, participant systems are required to return a textual output that satisfies the information needs of a given query. The type of the query is also given as one of 8 types: ARTIST, ACTOR, POLITICIAN, ATHLETE, FACILITY, GEO, DEFINITION, and QA. The limit length of the textual output depends on the type of output device, MOBILE and DESKTOP, and the limits are respectively 140 and 500. The output is referred to as X-string. The setting of input source has three types, Mandatory, Oracle and Open. Mandatory input source provides entire Web search results, and Oracle gives also a list of relevant Web pages for each query. In the setting of Open, participants generate X-strings using not only provided search results but also their own Web search results. The evaluation is based on nuggets, which is also used in 1CLICK task evaluation. A nugget is a text fragment that is relevant to the information need for a given query. In this task, nuggets are divided into iUnits, where their contents are independently relevant, atomic, and dependent. The evaluation of X-strings is based on matched iUnits and their position in the X-string. For more details on the NTCIR-10 1CLICK-2 task, we refer the reader to [1].

3. SUBMODULARITY

Submodularity is a property of a set function for a finite universe V. The submodularity in discrete optimization corresponds to convexity in continuous optimization. In contrast to *modularity* is a property of the functions that returns the sum of scores of the elements, the submodular function returns a value less then the sum. Many discrete optimization problems including the maximum coverage problem, and the max cut in graphs can be regarded as a submodular optimization problem.

Formally, the submodular function is defined as $f : 2^V \to \mathbb{R}$ that maps a subset $S \subseteq V$ to a real value. If for any $S, T \subseteq V$, $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$, then f is called *submodular*. The definition is equivalent to that of the *diminishing return* that is well known in the field of economics: $f(S \cup \{u\}) - f(S) \leq f(T \cup \{u\}) - f(T)$, where $T \subseteq S \subseteq V$ and u is an element of V. The diminishing return means that the value of element u remains the same or decreases as the set S becomes larger. The property is suitable for the summarization, because the gain to insert a new sentence to a summary that already contains sufficient information should be small. Therefore, many researches formalizing text summarization as submodular maximization have been done [7, 8, 11]. These approaches are, however, based on sentence extraction.

The concept of submodularity plays an important role in discrete optimization. The advantage of submodularity is some types of submodular maximization problem can be optimized approximately by a greedy algorithm with performance guarantee. The performance guarantee $1 - e^{-1}$ is the best possible ratio for submodular maximization. For the maximization of monotone submodular functions under a size limit of the set have a greedy algorithm with performance guarantee $1 - e^{-1}$. For the budgeted maximization, Kulik [2] proved $\frac{1}{2}(1 - e^{-1})$ for monotone non-decreasing submodular maximization.

In the field of constrained maximization, Kulik et al. [4] proposed an algorithm that solves submodular maximization under multiple linear constraints with performance guarantee $1 - e^{-1}$ in polynomial time. Although the approach can represent more flexible constraints, we cannot use their algorithm to solve our problem, because the algorithm needs to enumerate numerous combinations of elements. Integer linear programming (ILP) formulations can also represent such flexible constraints, and is commonly used to model text summarization [9]. However, it is hard to exactly solve large-scale ILP problems in practical time.

4. A VARIATION OF QUERY SNOWBALL METHOD

In this paper, we address the problem of handling joint model of sentence compression and sentence extraction. Thus our system needs finer grained scoring of query relevant content than the sentence similarity, while off-the-shelf cosine similarity cannot be applied to word scoring. With this view, we employ an approach based on Query SnowBall (QSB) proposed by [11]. The QSB is a query relevance scoring method that is designed for query-oriented summarization and calculates query relevance of each word leveraging word co-occurrence in source documents of summarization. Figure 1 shows the concept of QSB.

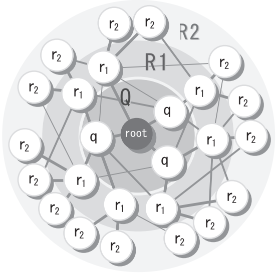

Figure 1: Co-occurrence Graph (Query Snowball): Q is the set of query terms (each represented by q), $R1$ is the set of words ($r1$) that co-occur with a query term in a sentence, and $R2$ is the set of words ($r2$) that co-occur with a word from $R1$, excluding those that are already in $R1$. The imaginary root node at the center represents the information need, and we assume that the need is propagated through this graph, where edges represent within-sentence co-occurrences. Thus, to compute sentence scores, we use not only the query terms but also the words in $R1$ and $R2$.

As shown in Figure 1, Q is a set of query terms, R1 is a set of words that co-occurred with Q on a sentence. R2 is a set of words that co-occur with R1, and not co-occur with Q. QSB aims to give a higher score to descriptions of a given query, so that the method allocates query relevant score to both words that co-occured with query terms, and words that indirectly co-occurred with them. Since the QSB focused on newspaper corpus, the method can not adapt to major change of source document size, and has a problem that the obtained score depends on the size of source documents. Therefore, the method allocates a large amount of scores to irrelevant word with query when the size of source document is too large. This is because, the method calculates relevance between words using word co-occurrences that is not normalized by the size of source documents. Thus, we modify query relevance score by normalizing the co-occurrences for the word frequency, so that the score becomes independent of the size of source documents.

We use frequency of uni-gram in Web Japanese N-gram [3] for calculating a score of informativeness of a word $s_b(w)$: $s_b(w) = \log(N/tf(w))$, where $tf(w)$ is the term frequency of w. We refer the score as base word score. Then, the query relevance score is calculated as a weight propagation from query terms to each node on the co-occurrence graph shown in Figure 1. The scores of words are represented as the product of the base word score and the degree of relevance with query terms. We describe how to compute the word scores for words in $R1$ and then those for words in $R2$.

As mentioned before, we introduce normalization based on the size of documents. In detail, we normalize co-occurrence frequency between words by the frequency of the word occurrences. Instead of the original definition of [11], we represent the weight propagation as follows:

$$\begin{aligned} s_r(r1) &= s_b(r1)\sum_{q\in Q}\left(\frac{s_b(q)}{sum_Q}\right)\left(\frac{\textit{co-occurrence}(q,r1)}{\textit{freq}(q)}\right), \\ sum_Q &= \sum_{q\in Q} s_b(q), \end{aligned}$$

where $\textit{co-occurrence}(w_1, w_2)$ is the frequency of the co-occurrence of two words w_1 and w_2 within a sentence in the source documents, and $\textit{freq}(w)$ is the term frequency of the word w in the source documents. The equation is expressed in the form of the products of the base word score and the strength of the relation between the word $r1$ and given query. The strength is also represented as the sum of the product of the two parts. The term $\left(\frac{s_b(q)}{sum_Q}\right)$ represents the presence of the query term q on query terms. The next term $\left(\frac{\textit{co-occurrence}(q,r1)}{\textit{freq}(q)}\right)$ indicates the ratio of co-occurrence between the word q and $r1$ to the number of sentences in which the word q occurred.

Similarly, the query relevance score for $r2 \in R2$ is computed based on the base word score of $r2$ and the relationship between $r2$ and $r1 \in R1$:

$$\begin{aligned} s_r(r2) &= \sum_{r1\in R1} s_b(r2)\left(\frac{s_r(r1)}{sum_{R1}}\right)\left(\frac{\textit{co-occurrence}(r1,r2)}{\textit{freq}(r1)}\right), \\ sum_{R1} &= \sum_{r1\in R1} s_r(r1). \end{aligned}$$

The scores for $r2 \in R2$ is also represented as the product of base score and the strength of the relation between the word $r2$ and query. The difference of the equation from that of $R1$ is that the score is based on the score of calculated score instead of base scores.

5. SUBTREE EXTRACTIVE SUMMARIZATION

We adopt the approach proposed by [10]. In order to handle the joint model of sentence compression and sentence extraction, we cast the model as a single process that extracts subtrees of dependency trees of sentences. We regard a sentence as a set of subtrees as shown in Figure 2. We refer the set as a bin.

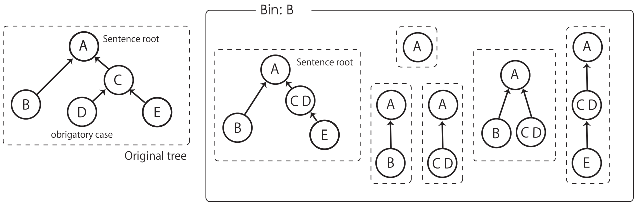

Figure 2: Valid subtrees in a bin.

Then, the model extracts a subtree as a compressed sentence. In Japanese, most of the dependency edges can be pruned without loss of grammaticality as long as the subtree keeps the root of dependency tree, as shown in Figure 3. However, the sentence generated by a subtree can be unnatural even though the subtree contains the root node of the sentence as shown (c) in Figure 3. In order to avoid generating such unnatural sentences, we need to detect and retain the obligatory dependency relations in the dependency tree. We address the problem by imposing must-link constraints if a phrase corresponds to an obligatory case of the main predicate. We merge obligatory phrases with the predicate beforehand so that the merged nodes make a single large node, as shown in Figure 2. We refer the subtrees that satisfy those constraints as *valid*. For dependency parsing, we used KNP [6]. Since KNP has internally a flag that indicates the case is "obligatory case" or "adjacent case," we regarded dependency relations flagged by KNP as obligatory dependency relations in the sentence compression.

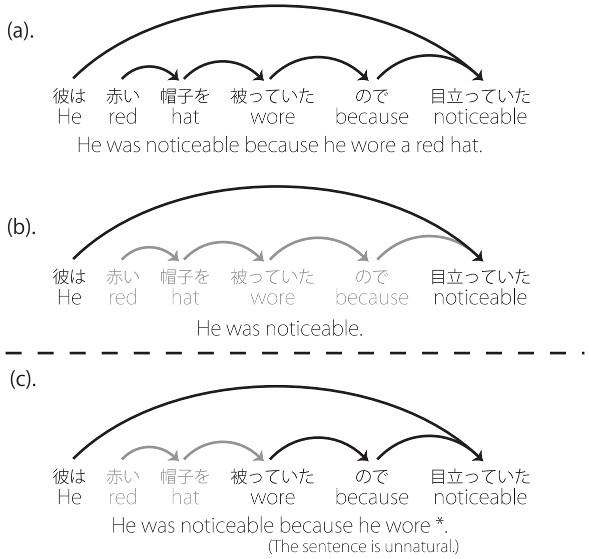
(a).
彼は 赤い 帽子を 被っていた ので 目立っていた
He red hat wore because noticeable
He was noticeable because he wore a red hat.
(b).
彼は 赤い 帽子を 被っていた ので 目立っていた
He red hat wore because noticeable
He was noticeable.
(c).
彼は 赤い 帽子を 被っていた ので 目立っていた
He red hat wore because noticeable
He was noticeable because he wore *.
(The sentence is unnatural.)

Figure 3: An example of how we can omit words in Japanese sentence.

We formalize the subtree extractive summarization as a submodular maximization problem. Let d be the damping rate, $count_S(w)$ be the number of sentences containing word w in summary S, $words(S)$ be the set of words included in summary S, $qsb(w)$ be the query relevance score of word w, and γ be a parameter that adjusts the rate of sentence compression. Our objective function for a summary S is represented as follows:

$$\begin{aligned} f(S) &= \sum_{w\in words(S)}\left\{\sum_{i=0}^{count_S(w)-1} qsb(w)d^i\right\} + \gamma reward(S) \\ reward(S) &= c(S) - |S|. \end{aligned}$$

We generate a summary by solving the maximization problem of the objective function under the constraint that the summary S consists of subtrees containing root nodes of dependency trees of sentences. It is proved that the problem can be approximately solved by greedy selection of the *maximal density subtree* that has the largest cost per its length by [10]. The maximal density subtree of each sentence can be obtained using a dynamic programming approach.

6. EXPERIMENTS

In this section, we report our results on oracle (ORCL) runs and a MANDATORY run in NTCIR 10 1CLICK-2. We

run name	ARTIST	ACTOR	POLITICIAN	ATHLETE	FACILITY	GEO	DEFINITION	QA	MEAN
ORG-J-D-ORCL-3	**0.263**	**0.293**	**0.317**	**0.306**	0.097	0.069	**0.294**	0.127	**0.206**
TTOKU-J-M-ORCL-1	0.132	0.160	0.096	0.109	0.060	**0.177**	0.036	**0.223**	0.124
TTOKU-J-D-ORCL-2	0.076	0.133	0.098	0.081	0.086	0.075	0.026	0.098	0.082
TTOKU-J-M-MAND-3	0.019	0.071	0.043	0.017	0.000	0.054	0.018	0.092	0.040
The best score of other runs	0.149	0.165	0.138	0.230	**0.224**	0.106	0.154	0.098	0.117

Table 1: S#-measure of ORCL runs for each query type.

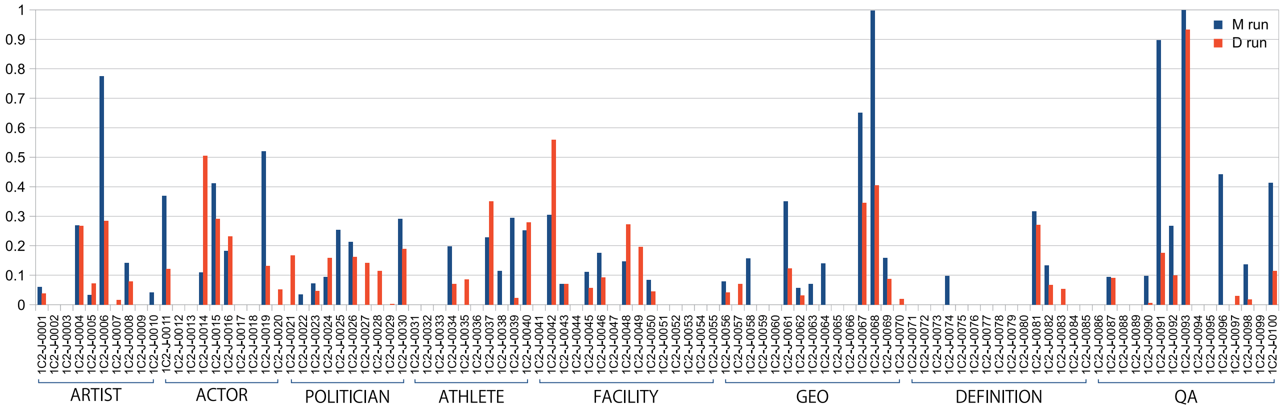

Figure 4: S#-measure(I) of TTOKU-J-ORCL results for each query.

submitted a Desktop run and a Mobile run on ORCL setting, and one Mobile runs for MANDATORY setting. We utilized JUMAN [5] for morphological analysis, and used KNP [6] for dependency parsing. We used ACLIA1 dataset for the parameter tuning. We did not employ any specialization strategy for types of queries, and types of run settings.

6.1 Evaluation

In 1CLICK-2, S#-measure is employed as the primary metric. The measure is harmonic mean of S-measure and T-measure, both of measures are based on iUnit matching. The S-measure is an extended weighted recall that takes into account the position of matches. The T-measure is a precision like metric that denotes a fraction of the length of vital strings of the length of X-strings. The S#-measure plays a role like F-measure. For more details, please refer to [1]. Systems are evaluated by S#-measure of each run.

6.2 Results

We describe the results of our submitted runs. Table 1 summarizes official results of our runs and other runs of participants in 1-CLICK2. ORG-J-D-ORCL-3 is submitted by organizers and described as “top-ranked snippets.” TTOKU-J-M-ORCL-1, TTOKU-J-D-ORCL-2, and TTOKU-J-M-MAND-3 are our submitted runs. The bottom line indicates the best score for each query of other participants.

Our mobile run (TTOKU-J-M-ORCL-1) achieved the best mean S#-measure over runs of participants. Since our runs are oracle runs, our runs are of advantage in comparison. However, in the GEO and QA queries we have the best score over every run (including other ORCL runs). Figure 4 shows the results of our ORCL runs for each query. The scores of GEO and QA are driven up by a few queries, 1C2-J-0067, 1C2-J-0068, and 1C2-J-0093 and so on. For example, in the result of query 1C2-J-0068 (Kawaguchi-city police station), our system extracts lines that enumerate names of police station. The system gives scores to words that co-occurred with query terms, the names of police station have higher score than the other terms, because the names co-occur with query term “交番 (police station)” frequently. In addition, only few other words co-occurred with the “police station” and “郵便局 (post office)” in 1C2-J-0067. In the results of 1C2-J-0093 (Why is the sky blue), a query term “青い (blue)” frequently co-occurred with “分子 (molecule),” “散乱 (scattering),” and other important terms that are used in the description of the reason why the sky is blue.

Our MANDATORY run (TTOKU-J-M-MAND-3) is the worst over every run. Since our system does not consider the rank of retrieved documents, our system simply generates summaries of all retrieved documents. Moreover, Query Snowball tends to give scores to irrelevant terms when the source documents include irrelevant documents. That causes the low score of our MANDATORY run.

	S#	S	T	Weighted recall
Mobile	0.124	0.127	0.107	0.049
Desktop	0.082	0.083	0.050	0.059

Table 2: Mean S#-measure, S-measure and T-measure, weighted recall of our runs. (L=500)

Table 2 shows mean S#, S, T-measure and weighted recall of our runs. Our Mobile run outperforms our ORCL runs in all metrics except for weighted recall. In spite of our Desktop run has more than twice length limit, there are only few

(A)

Original:
パーキンソン病 (パーキンソンびょう、英: Parkinson’s disease) は、脳内のドーパミン不足と
アセチルコリンの相対的増加とを病態とし、錐体外路系徴候 (錐体外路症状) を示す疾患である。
Parkinson’s disease (En: Parkinson’s disease) is a disease that has clinical conditions of insufficient dopamin in brain
and relative increase of acetylcholine, and has extrapyramidal symptoms.

Compressed:
病態とし、錐体外路系徴候 (錐体外路症状) を示す疾患である。
Clinical conditions of ..., it has extrapyramidal symptoms

Figure 5: Examples of compressed sentences for the query “Parkinson’s disease.”

(B)

Original:
英語で「vanilla sky」(バニラ・スカイ) と呼ばれる状態の空
A sky that is called as vanilla sky in English.

Compressed:
空
Sky.

(C)

Original:
空は　なぜ青いの?
Why is the sky blue?

Compressed:
Compressed: なぜ青いの?
Why is it blue?

Figure 6: Examples of compressed sentences for the query “Why is the sky blue?”

improvements in weighted recall. The result shows the outputs after 140 characters are not less informative. Since our algorithm tends to extract contents that are allocated high scores at first, the results also suggests our score function is not reliable when it returns relatively low values. Although our mobile have not higher weighted recall than that of other runs, our mobile run has high S# and S-measure. That suggests informative contents tend to appear in former position of X-strings, since sentence compression reduce the length of each description.

6.3 Error analyses of sentence compression

Figures 5 and 6 show examples of compressed sentences in our methods. (A) consists of both a successful example and an example of error on compression at the same time. The sentence includes three iUnits: insufficient dopamine, increase of acetylcholine, and extrapyramidal symptom. When a summary already contains the query “Parkinson’s disease,” users may know the topic of the summary. It is not beneficial to contain Parkinson’s disease again. Therefore the compression is expected to drop the first clause that describes the name of Parkinson’s disease. However, the compression also drops most parts of the second clause that contains 2 iUnits: insufficient dopamine and increase of acetylcholine. The drop of necessary part of clause makes the second clause to be unnatural, because the compressed sentence lacks the content of “clinical condition.” In this case, obligatory dependency should work for saving the 2 iUnits. Our sentence compression does not cover the case.

Other examples of errors on compression are (B) and (C). In those examples, compression extracts only query terms from documents. This kind of errors are noticeable in the beginning of the summary, because the query term itself tends to have the highest score per its length. The extracted sentence is not so unnatural, but short. The compression method still needs more improvement to avoid generating too short sentences.

7. CONCLUSIONS

In this paper, we proposed a new query-oriented summarization method and applied the method to NTCIR-10 1CLICK-2 tasks. We refined the methods used in NTCIR-9 1CLICK for variable size documents. Experimental results show that we have achieved best score in GEO and QA query types.

8. REFERENCES

[1] M. P. Kato, M. Ekstrand-Abueg, V. Pavlu, T. Sakai, T. Yamamoto, and M. Iwata. Overview of the NTCIR-10 1CLICK-2 Task. In *Proceedings of the 10th NTCIR Workshop*, NTCIR-10. NTCIR, 2013.

[2] A. Krause and C. Guestrin. A note on the budgeted maximization on submodular functions. Technical Report CMU-CALD-05-103, Carnegie Mellon University, 2005.

[3] T. Kudo and H. Kazawa. *Web Japanese N-gram Version 1*. Gengo Shigen Kyokai, 2007.

[4] A. Kulik, H. Shachnai, and T. Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’09, pages 545–554, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[5] S. Kurohashi and D. Kawahara. *Japanese Morphological Analysis System JUMAN 6.0 Users Manual*, 2009. `http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN`.

[6] S. Kurohashi and D. Kawahara. *KN parser (Kurohashi-Nagao parser) 3.0 Users Manual*, 2009. `http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP`.

[7] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT ’10, pages 912–920, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[8] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for*

Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pages 510–520, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[9] R. McDonald. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European conference on IR research*, ECIR'07, pages 557–564, Berlin, Heidelberg, 2007. Springer-Verlag.

[10] H. Morita, S. Ryohei, T. Hiroya, and O. Manabu. Subtree extractive summarization via submodular maximization. In *Proceedings of 51st Annual Meeting of the Association for Computational Linguistics (to appear)*. Association for Computational Linguistics, 2013.

[11] H. Morita, T. Sakai, and M. Okumura. Query snowball: a co-occurrence-based approach to multi-document summarization for question answering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 223–229, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[12] J. Tang, L. Yao, and D. Chen. Multi-topic based query-oriented summarization. In *Proceedings of 2009 SIAM International Conference Data Mining (SDM'2009)*, pages 1147–1158, 2009.