Description of the NTOU MobileClick System at NTCIR-11

Chi-Ting Liu and Chuan-Jie Lin
Department of Computer Science and Engineering
National Taiwan Ocean University
2 Pei-Ning Road, Keelung, Taiwan 20224
{ct.liu,cjlin}@ntou.edu.tw

ABSTRACT

This paper describes the design of NTOU's first MobileClick system participating in two NTCIR-11 MobileClick English subasks, iUnit Retrieval and iUnit Summarization. Our iUnit retrieval module first used inverted query frequency (**iqf**) to extract topic-related keywords, and then identify important nuggets by measuring and sorting **nf·iqf** scores, where **nf** is nugget frequency. Summarization module is a greedy clustering system according to the lengths and sizes of common leading substrings among iUnits. Our iUnit Retrieval formal run did not perform well, where nDCG@10 score is 0.1426 and Q@10 is 0.0803. But our iUnit Summarization formal run was ranked at the first place, where M-measure score is 4.43 at the patience parameter L=280.

Team Name

NTOUA

Subtasks

iUnit Retrieval (English)
iUnit Summarization (English)

Keywords

iUnit retrieval, summarization, topic-related keyword, nugget frequency, query frequency, longest common leading substring

1. INTRODUCTION

MobileClick [3] is interested in how to display IR results on a length-limited device, such as a mobile phone. There have been two preceding related tasks held in NTCIR, 1Click tasks [2, 7].

There are two subtasks in MobileClick Task. The main goal of iUnit retrieval is to retrieve novel and important information from relevant documents. Although it is very similar to multi-document summarization [1, 4, 5, 6], the task does not generate a summary directly, but extract informative units (iUnits) and then use them to generate length-restricted, hierarchical summaries, which becomes the main goal of another subtask, iUnit Summarization.

It is our first time to participate in the MobileClick tasks. Due to the lack of time for system developing, we only proposed one model for each subtask.

We organize the paper as follows. In Section 2 we overview the strategies leading us to a efficient passage retrieval framework. In Section 3 we describe our summarization system, and in Section 4 we discuss the experiment results. We conclude this paper in Section 5.

2. IUNIT RETRIEVAL

2.1 Identifying Topic-Related Keywords

Our basic assumption is that a nugget containing important information should often carry topic-related keywords. A ***topic-related keyword*** is a word carrying important information about a specific topic (query). We use ***nugget frequency*** and ***inverted query frequency*** to capture such keywords. Nugget frequency and inverted query frequency are defined in the similar way as term frequency (tf) and inverted document frequency (idf).

Nugget frequency $nf(w, q)$ is the number of nuggets relevant to a query q that contains a word w. Query frequency $qf(w)$ counts the number of queries that at least one of their relevant nuggets contains a word w. Similar to document frequency, a more discriminative word has less query frequency, so inverted query frequency is used instead:

$$iqf(w) = \log \frac{N}{qf(w)} \tag{1}$$

where N is the total number of queries in the dataset.

However, topic-related keywords should not be too infrequent. We hence ignored those words with nugget frequency less than 3. Moreover, stop words were not considered. Long words with a length larger than 255 characters were also discarded (if any).

Relevant documents were first segmented into nuggets by six punctuation marks, comma (,), period (.), exclamation mark (!), question mark (?), colon (:), and semicolon (;). All words in queries or nuggets were lemmatized. Only nuggets containing any of its corresponding query terms were considered as ***relevant nuggets***.

For each query, each distinct word w in its relevant nuggets were sorted according to their $nf{\cdot}iqf$ scores. Top 30 words were selected as topic-related keywords related to this query. Table 1 lists some examples of topic-related keywords selected from the formal test.

2.2 Selecting Representative Nuggets

Let Q be the set of queries and $G(q)$ be the set of relevant nuggets to a query $q \in Q$. For each nugget $g \in G(q)$, its ***information-containing score*** is defined as:

$$\text{info\_score}(g) = \sum_{w \in g} nf(w, q) \times iqf(w) \tag{2}$$

Table 1: Examples of Topic-Related Keywords.

ID	Query	Keyword
Good Performance		
MC-E-0003	why does turkey make you sleepy	tryptophan, thanksgiving, health, amino, alcohol, l-tryptophan, blood, actually, caffeine, serotonin, fitness, rss, drowsy, enough, entire ...
MC-E-0024	why do we yawn	contagious, oxygen, carbon, social, ago, excessive, people, dioxide, involuntary, brain, different, thinking, empathy, reflex, actually, contagion ...
MC-E-0036	how is trash processed	garbage, waste, management, compactors, disposal, empty, landfill, municipal, compactor, energy, news, recovery, recyclables, electricity, information, environmental, plasma, residential, solid, plastic, china, industrial ...
Bad Performance		
MC-E-0001	java vs python text processing	perl, ruby, c++, php, xml, c#, api, microsoft, same, syntax, software, natural, language, first, better, web, really, two, available ...
MC-E-0002	hiphop clubs barcelona	luxembourg, del, fc, en, music, el, madrid, fiesta, north, hop-on, tour, south, los, dj, de, es ...
MC-E-0008	best computer working position	online, knowledge, information, available, ergonomic, internet, software, full-time, management, different, health, ago, entertainment, medical, home, employment, important, technical ...

where w is a topic-related keyword that occurs in g. This score will be further normalized by the information-containing score of the whole nugget set.

$$\text{info\_score}^{\star}(g) = \frac{\text{info\_score}(g)}{\sum_{w \in G(q)} nf(w,q) \times iqf(w)} \quad (3)$$

Because nuggets are often long sentences where keywords only appear in a shorter window, we trim each nugget to its longest span that covers all topic-related keywords and query words appearing in this nugget.

Nuggets were sorted by their information-containing scores. An unselected nugget g with the largest score was selected as an iUnit if it was not too long (> 70 bytes) or too similar to any selected iUnits. Dice coefficient was used to measure nugget similarity and the threshold was set to be 0.6. Maximally 50 nuggets would be output as iUnits to one query.

3. IUNIT SUMMARIZATION

The task of iUnit summarization is to identify important information and make a short summarization fit in a screen, while second important information is also provided but only with headlines (or "links" as in task definition) where interested users can click and read in a second screen. Approaches to produce two-layer texts and second-layer links are described in the following subsections.

3.1 Constructing First Layer Content

Our approach is greedy clustering based on longest common leading substrings. iUnits having the same leading strings can be merged into one sentence by concatenating remaining parts of these sentences into this common leading substring. Let $\text{CLS}(u_i, u_j)$ be the longest common leading substring of the iUnits u_i and u_j. Algorithm of clustering and text generation is as follows.

1. For each pair of iUnits not yes selected into final summarization, find their longest common leading substring.
2. Let $\text{CLS}(t_i, t_j)$ be the longest string among all the longest common leading substrings, collect all the iUnits u_k having the same leading substring as $\text{CLS}(t_i, t_j)$ to become a cluster.
3. A merged sentence like

   $$\text{CLS}(t_i, t_j)\{u_1 - \text{CLS}(t_i, t_j)\}, \{u_2 - \text{CLS}(t_i, t_j)\}, ...$$

   is generated and merged into summary.
4. Repeat these steps until
   (a) The length of the summary meets the length restriction, or
   (b) No remaining iUnits having common leading substrings.

Take Topic MC-E-0001 as an example. The organizers provided the following nuggets:

u_1 Java documentation is extensive
u_2 Python is more expressive
u_3 Java is more verbose
...
u_{12} Python can be written more quickly
u_{13} Python can be maintained more easily
u_{14} Natural Language Toolkit (NLTK) provides a lot of tools for natural language processing
u_{15} Python is a more natural language
...

After clustering by longest common leading substrings, the following clusters are generated and sorted according to the lengths of CLS.

C_1 :

[python can be] written more quickly
[python can be] maintained more easily

C_2 :

[python is a] more natural language
[python is a] dynamically-typed language allowing higher productivity

...

By concatenating non-overlapping parts of iUnits with the longest common leading substring, the following summary sentences are generated.

S_1 : python can be written more quickly, maintained more easily

S_2 : python is a more natural language, dynamically-typed language allowing higher productivity

Concatenate these summary sentences until length limit is reached (280 bytes in this example, so three more sentences are included.) The resulting text will be shown in the first layer.

S_3 : python has clear concise syntax, extensive libraries

S_4 : python is more expressive, easier to learn, difficult for beginners

S_5 : java is more verbose, easy, faster

The method is simple and automatic. The readibility of the resulting sentences is not bad. We do not know any other system using similar method to merge several sentences into one.

3.2 Constructing Second Layer Content

Second layer collects the iUnit clusters which do not appear in the first layer. These clusters are re-sorted according to the number of iUnits in the clusters. The first two largest clusters are selected to become content in the second layer. Their common leading substrings become anchor text of the links in the first-layer page.

The choice of at most "two" second-layer clusters is heuristic. We think that a mobile user may not want to see too many information in a screen. Giving too many links will fall back into the "tem-blue-link" paradigm. In the future, if we have more experimental data, this number can be decided more automatically.

Content in the second layer pages is prepared in the same way as the first-layer page. Sentences in each cluster are concatenated until length limit is reached.

Take Topic MC-E-0001 as an example again. The two largest clusters not appearing in the first layer are:

C_6 :

u_1 [java] documentation is extensive

u_9 [java] has weird syntax

u_{17} [java]

C_7 :

u_{11} [natural language] processing

u_{14} [natural language] toolkit (NLTK) provides a lot of tools for natural language processing

...

So we provide two links, "java" and "natural language", linking to their corresponding second-layer summary sentences, respectively.

Table 2: Performance of NTOU iUnit Retrieval formal run.

Metric	NTOU
nDCG@5	0.1529
nDCG@10	0.1426
nDCG@80	0.0927
nDCG@400	0.0841
Q@5	0.1063
Q@10	0.0803
Q@80	0.0222
Q@400	0.0179

Table 3: M-measures of NTOU iUnit Summarization formal run (comparing to gold standard).

	NTOU	Gold
Patience parameter		
L=140	1.91	6.33
L=280	4.43	8.59
L=560	8.14	15.4
L=840	9.84	17.8
Query Category		
ALL QUERIES	9.84	17.8
LOOKUPLIST	14.7	21.6
FACTFINDING	6.55	15.8
COMPARISON	10.3	16.9

4. FORMAL RUN RESULTS

4.1 Evaluation Results

This year, due to time limitation, we only designed one model to do iUnit retrieval and one model for iUnit summarization. We submitted one English formal run to iUnit Retrieval Mandatory subtask and one English formal run to iUnit Summarization Mandatory subtask, respectively.

Main evaluation metrics in the iUnit Retrieval task are nDCG@k and Q-measures. The nDCG@10 score of our formal run is 0.1426 and the Q@10 is 0.0803, which is not very good. More details are listed in Table 2.

Main evaluation metrics in the iUnit Summarization task are M-measures. Our run was ranked at the first place among the four submitted runs in all levels and all categories of queries. But comparing to the upper bound, our run achieved only half of the scores measured on gold standard. There is still space to be improved. More evaluation results are listed in Table 3.

4.2 Discussion

Performance of our iUnit retrieval formal run was not satisfying. We think that the way to identify topic-related keywords is not suitable to every kind of information need.

In Table 1, three examples are topics receiving good IR results and three receiving bad results. As we can see, if real important keywords were successfully identified, such as "tryptophan" for Query003 and "oxygen" for Query024, retrieval performance would be better.

But observing topic-related keywords for Query001 and Query008, we find that keywords are still greatly related to the topics. However, they are not key points to fulfill information needs. Keywords for Query001 are mostly terms about programming languages, not features comparing two

programming languages. Keywords for Query008 are mostly IT terms, not working positions.

Keywords for Query002 reveal another problem. Most of them are Spanish stopwords, because many names of Spanish hiphop clubs contain Spanish function words. It makes our system easily to extract Spanish sentences from the relevant documents, but the sentences do not necessary contain any name of hiphiop clubs.

5. CONCLUSIONS

This paper describes the systems of NTOU's first attempt to participate in two NTCIR-11 MobileClick English subasks, iUnit Retrieval and iUnit Summarization.

The iUnit Retrieval system identified topic-related keywords with high nugget frequency (**nf**) and inverted query frequency (**iqf**) scores, and then selected nuggets containing more of these keywords as iUnits. Our iUnit Retrieval formal run achieved a nDCG@10 score at 0.1426 and Q@10 at 0.0803, which was not very good.

The iUnit Summarization system clustered iUnits by longest common leading substrings with a greedy clustering algorithm. The top-ranked clusters were used to construct the content of the first layer and the remaining clusters were used to construct the links and content of the second layer. Our iUnit Summarization formal run was ranked at the first place, where M-measure score is 4.43 at the patience parameter L=280. But comparing to the upper bound, our run achieved only half of the scores measured on gold standard. There is still space to be improved.

6. REFERENCES

[1] D. Das and A. F. Martins. A Survey on Automatic Text Summarization. Technical report, Literature Survey for the Language and Statistics II course at Carnegie Mellon University, 2007.

[2] 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 Meeting on Evaluation of Information Access Technologies (NTCIR-10)*, pages 182–211, Tokyo, Japan, 2013.

[3] M. P. Kato, M. Ekstrand-Abueg, V. Pavlu, T. Sakai, T. Yamamoto, and M. Iwata. Overview of the NTCIR-11 MobileClick Task. In *Proceedings of the 11th NTCIR Workshop Meeting on Evaluation of Information Access Technologies (NTCIR-11)*, Tokyo, Japan, 2014. to be appeared.

[4] J.-J. Kuo, H.-C. Wung, C.-J. Lin, and H.-H. Chen. Multi-document Summarization Using Informative Words and Its Evaluation with a QA System. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2002)*, pages 391–401, Mexico City, Mexico, 2002.

[5] Y. Ouyang, W. Li, S. Li, and Q. Lu. Applying regression models to query-focused multi-document summarization. *Information Processing and Management*, 47(2):227–237, 2011.

[6] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, 4(6):919–938, 2004.

[7] T. Sakai, M. P. Kato, and Y.-I. Song. Overview of NTCIR-9 1CLICK. In *Proceedings of the 9th NTCIR Workshop Meeting on Evaluation of Information Access Technologies (NTCIR-9)*, pages 180–201, Tokyo, Japan, 2011.