CUIS at the NTCIR-12 MobileClick2 Task

Kwun Ping Lai
Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong
kplai@se.cuhk.edu.hk

Wai Lam
Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong
wlam@se.cuhk.edu.hk

Lidong Bing
Machine Learning Department,
Carnegie Mellon University
lbing@cs.cmu.edu

ABSTRACT

We present our approach for tackling the iUnit ranking and iUnit summarization subtasks of MobileClick2. We first conduct intent discovery based on latent topic modeling. Our iUnit ranking method exploits the discovered intents and considers the importance of an iUnit in each Web content document. We further develop our iUnit summarization model using the outcome from the iUnit ranking subtask. Our result submitted to the iUnit ranking subtask demonstrates superior performance.

Team Name

CUIS

Subtasks

iUnit ranking subtask (English), iUnit summarization subtask (English)

Keywords

topic modeling, intent discovery, Web content extraction

1. INTRODUCTION

MobileClick2 task [5], organized by NTCIR-12, aims at solving information retrieval problem specifically in mobile platform. Returning a list of links relevant to the query is not suitable because of two reasons: 1) the small screen size of mobile platform is not suitable for displaying large amount of information and 2) the slow network speed and computing power of mobile platform greatly increase the time burden of users visiting the links one by one to locate the final desired information. The two subtasks, namely, iUnit ranking and iUnit summarization address the two problems stated above. The iUnit ranking subtask focuses on ordering the iUnits (information unit). Displaying only the top ranked iUnits can solve the small screen size problem. The iUnit summarization subtask requires a two-layered structure with a brief summary and link of intents (interpretation) in the first layer and detailed description of different intents in the second layer. Users can easily visit a particular interested intent.

The input of the iUnit ranking subtask is a query, a list of iUnits and a collection of retrieved HTML documents for the query. For the summarization subtask, the input is similar except for an additional list of intent labels.

The rest of the paper is organized as follows. We first present the related works in Section 2. We then introduce our approaches used to handle the subtasks in Section 3. In Section 4, we present the experimental results. We finally conclude our paper in Section 5.

2. RELATED WORKS

MobileClick2 is a continuous investigation of the task MobileClick. The iUnit ranking and iUnit summarization subtasks were partially and fully launched in the previous MobileClick respectively. The previous iUnit retrieval subtask required to find out the iUnits automatically.

One promising model for the iUnit retrieval subtask is IISR [2]. It applies topic-based scoring method to sentence ranking which can be used to rank iUnits. However, the SearchRank score of the model directly utilizes the ranking from the search engine that may not be entirely suitable. In our approach, we re-rank each page by the discovered hidden topics to support the iUnit ranking process.

For the iUnit summarization subtask, the baseline method of organizers [4] demonstrates good performance. It puts the remaining iUnits into the second layer according to the similarity between the iUnit itself and the header plus the following text of the page. However, this strategy ignores the remaining documents. Our approach takes all documents into account.

3. OUR APPROACH

3.1 Overview

Each query is associated with a set of top ranked 300-500 Web documents retrieved using a search engine. Raw documents may generally contain semi-structured content such as HTML tags, scripts, styles, comments etc. In our proposed framework, the first major component is a Web content extraction component which extracts the main content from a Web document. We investigate two methods, namely, tag-based method and text feature-based method. After the main content is extracted from each raw Web document, we convert the text to lower case to obtain a collection of Web content documents. The next step is to conduct hidden topic discovery modeling intents from the collection of Web content documents. We make use of the discovered intents to rank all iUnits for the iUnit ranking subtask. Using the iUnit ranking result, we intend to develop a iUnit summarization model. However, due to limited time, we can only develop a simplified model.

3.2 Web Content Extraction

All components inside a HTML document can be divided into two groups: 1) visible component and 2) invisible component. Visible component contains visible information such as text which can be rendered so that users can see it on the screen. However, it is clear that not all visible components, such as navigation bars, contribute to the main content. Invisible components refers to the code that is not supposed to be rendered. Take the following HTML DOM node as example:

<h3 class='panel-title'>What is the task?</h3>

It is clear that only the words in between the tags is a visible component. We aim at extracting useful Web content from the visible components.

3.2.1 Tag-Based

We consider those HTML tags with which useful content is associated. The tags include paragraph (p tag), title (h tag), cell in table (td tag) and list (li tag). The text inside these tags is extracted and is saved to a plain text document.

We use Jsoup[1] Java library to achieve the implementation. It has a built-in selector that automatically extracts all DOM nodes with a given tag name. Finally, the inner-texts of the selected DOM nodes are extracted and are treated as the main content.

3.2.2 Text Feature-Based

We also investigate a second Web content extraction method which is based on a text feature-based method known as Boilerpipe[2] developed by Kohlschutter et al.[6]. This method exploits shallow text features to clean boilerplate. It makes use of the number of words and link density to distinguish the actual content from boilerplate text. Moreover, the computational cost of this model is small.

3.3 iUnit Ranking Model

3.3.1 Motivation

The purpose of iUnit ranking is to find out important iUnits. As stated in MobileClick2 task specification[3], the evaluation metric of an iUnit in this task is given as follows:

$$GG(u) = \sum_{i \in I_q} P(i|q) g_i(u) \tag{1}$$

where $GG(u)$ is the global importance of the iUnit u. $P(i|q)$ is the intent probability of the intent i given the query q. I_q is the intent set of the query q. $g_i(u)$ is the per-intent importance of the iUnit u. This metric illustrates that intents play an important role in the determination of iUnit importance. However, the intents are not known in advance. To obtain the global importance, a model should be capable of estimating the intent probability per query and also the iUnit importance per intent.

We develop our model by exploiting the hidden topics modeling intents discovered from the collection of Web content documents. Our model re-ranks the documents by the discovered intents and determines the importance of iUnits based on the discovered intents.

3.3.2 Intent Discovery

Our intent discovery model makes use of Latent Dirichlet Allocation (LDA) [1] which can be used to discover latent topics. We treat the latent topics as intents and we use these two terms interchangeably. Our intent discovery approach is inspired by the work of He et al.[3] which finds facets for search result diversification.

We first train a LDA model using the collection of Web content documents. We use Mallet [7] package for running the LDA model. The next step is to rank the set of intents based on the query likelihood. We treat the query as a short document and use the trained LDA model to get the inferred latent topic distribution. The weighting of each word of the query to each latent topic is inferred by the trained LDA model. By randomly sampling the latent topics for each word in the query, we get a distribution of the latent topics whose normalized form represents the latent topic distribution of the query. Precisely, suppose query q consists of r terms $\{q_1, q_2, ..., q_r\}$. For each term q_j, we conduct sampling for latent topic i based on the latent topic representation $P(i|q_j)$ calculated as follows:

$$P(i|q_j) = \frac{P(q_j|i)P(i)}{P(q_j)} \tag{2}$$

By setting a uniform prior, $P(i|q_j)$ is proportional to $P(q_j|i)$ which is inferred by the trained LDA model. We sample n times for each term q_j to get a total of nr latent topics. Finally, we obtain the probability $P(i|q)$ of the latent topic i given the query q formulated as follows:

$$P(i|q) = \frac{c_i}{nr} \tag{3}$$

where c_i is the number of latent topic i being sampled.

Then the latent topics are ranked by $P(i|q)$. This creates an intent ranking R_I which is used for Web document reranking described below.

We attempt to associate one intent for each Web document. The collection of Web documents per intent serves as an information pool to convey the importance of the iUnits corresponding to that intent. Each Web document d is assigned to an intent i^* in which the document has the highest topic probability formulated as follows:

$$i^* = \arg\max_{i \in I} P(i|d) \tag{4}$$

where I is the set of latent topics. The Web documents with the same intent i together form the intent document group G_i. We maintain the original ranking of Web documents within a group.

The documents are re-ranked using a round-robin manner strategy. At each round, the top Web document of all intent document groups is picked according to the ordering in R_I. The procedure is stated in Algorithm 1. The new document raking score is captured by $ranking(.)$. The top Web document will have a ranking score of 1 while the Web document ranked at the bottom will have $|D|$ which is the total number of documents. The $poll()$ function in Algorithm 1 removes the first element of a queue and returns it. In each round, the first documents of all intent document groups are removed and inserted into the new ranking list.

The new document ranking score of all Web documents will be used in formulating the intent-based document weight

[1] http://jsoup.org/

[2] http://www.l3s.de/~kohlschuetter/boilerplate/

[3] http://www.mobileclick.org/home/task

Algorithm 1 Web document re-ranking

```
function RR-RANK($|D|, G, R_I$)
input: The total number of Web document $|D|$; intent document group $G$; intent ranking $R_I$
output: New intent-based $ranking$ of all Web document
    $ranking = \emptyset$
    while $|ranking| \neq |D|$ do
        for each $i \in R_I$ do
            if $G_i \neq \emptyset$ then
                $ranking.add(G_i.poll())$
            end if
        end for
    end while
    return $ranking$
end function
```

w_d for each document d defined as follows:

$$w_d = |D| + 1 - ranking(d) \tag{5}$$

Another component is to determine the importance of an iUnit in a Web document. We design an importance measure $I(u, d)$ to capture the importance of the iUnit u in the Web document d as follows:

$$I(u, d) = \frac{\sum_{w \in \mathbf{u}} exist(w, d)}{|\mathbf{u}|} \tag{6}$$

where $\mathbf{u}$ is the set of words of the iUnit u excluding the stop words. The function $exist(w, d)$ is an indicator function denoting the existence of the word w in the Web document d.

$$exist(w, d) = \begin{cases} 1 & w \in d \\ 0 & w \notin d \end{cases}$$

We formulate the iUnit scoring as follows:

$$score(u) = \sum_{d \in D_q} w_d I(u, d) \tag{7}$$

Finally, all iUnits are ranked after calculating their scores which are a weighted sum of the importance measure of the iUnit per Web document. They are ranked by sorting their scores in descending order.

3.4 iUnit Summarization Model

3.4.1 Motivation

The input of iUnit summarization subtask includes one additional information which is the set of intent labels per query. As stated in MobileClick2 task specification, the evaluation metric of this subtask is M-measure which is a weighted sum of U-measure of each trailtext. The U-measure of trailtext in terms of the intent i is the weighted sum of the importance of iUnit in the intent i that is given below:

$$U_i(t) = \frac{1}{N} \sum_{j=1}^{|t|} g_i(u_j) d(u_j) \tag{8}$$

where d is the position-based decay function between 0 and 1. The iUnit at the top has a lower decay while the iUnit at the bottom has a higher decay. As all iUnits in the second layer will only be read by users after scanning the first layer and the intent, the U-measure score of the second layer is heavily reduced. Based on this motivation as well as intents, we develop our iUnit summarization model. However, due to limited time, we can only develop a simplified model.

3.4.2 Model Description

The first component in our iUnit summarization model is to assign one intent label to each Web document to form the intent document group D. We conduct intent label keyword search on the collection of Web content documents. The Web document is assigned to an intent document group D_{i^*} of the intent i^* in which the Web doucment has the highest probability. It is formulated as follows:

$$i^* = \arg\max_{i \in I_q} P(i|d) \tag{9}$$

where I_q is the given set of intents of the query q. The probability of the intent i given a document d is defined as follows:

$$P(i|d) = \frac{f(i, d)}{\sum_{i' \in I_q} f(i', d)} \tag{10}$$

$f(i, d)$ is the importance of the intent i in the Web document d defined as follows:

$$f(i, d) = \min_{w \in i} tf(w, d) \tag{11}$$

where w is a word inside the intent i. $tf(w, d)$ is the term frequency of the word w in the Web document d.

For the remaining Web documents with no words of intent label inside the content, the term weight $tf\text{-}idf$ is adopted for calculating the cosine similarity between themselves and each Web document in each intent document group. The probability of a remaining Web document d' that belongs to the intent document group D_i is defined as follows:

$$P(i|d') = \frac{\sum_{d'' \in D_i} sim(\boldsymbol{v}_{d'}, \boldsymbol{v}_{d''})}{|D_i|} \tag{12}$$

where $sim(\boldsymbol{v}_{d'}, \boldsymbol{v}_{d''})$ is the cosine similarity between the document vectors $\boldsymbol{v}_{d'}$ and $\boldsymbol{v}_{d''}$, $\boldsymbol{v}_{d'}$ is the $tf\text{-}idf$ vector of the remaining Web document d' and $\boldsymbol{v}_{d''}$ is the $tf\text{-}idf$ vector of the Web document d'' which belongs to the intent document group D_i. The remaining Web document d' is assigned to the intent document group D_{i^*} of the intent i^* with the highest average similarity:

$$i^* = \arg\max_{i \in I_q} P(i|d') \tag{13}$$

Next, we assign one intent to each iUnit. We conduct keyword search of the words in the iUnit on the collection of Web content documents from each intent document group. Then we assign the iUnit the intent of the intent document group with the highest occurrence rate.

After that, we construct the first layer by concatenating iUnits based on the ranking obtained from the iUnit ranking subtask until reaching a length limit of 420. The intent label that links to the second layer of each intent is placed at tail. For the remaining iUnits, they are placed into the corresponding intent layer again in the order obtained in the iUnit ranking subtask.

4. EXPERIMENTS

4.1 iUnit Ranking

There are three parameters in our iUnit ranking model, namely, the number of training iterations of the LDA model, the number of latent topics, and the number of Web documents used. We set the number of training iterations to 1000. We set the number of latent topics to be 10. We used all raw HTML documents as inputs for both subtasks. With the above settings, using the tag-based method for extracting Web content achieves a score of 0.903. The score improves to 0.9042 using the text feature-based method while keeping the same settings. Both scores are slightly higher than the best baseline method which has a score of 0.8975.

4.2 iUnit Summarization

The score of our iUnit summarization model is 16.4195. It is close to the best baseline method. We also investigate another run by removing tf-idf feature used to group the unidentified Web content documents. The score is 15.0659 showing a large performance reduction. It demonstrates that tf-idf is an important component of the model that considers term weights.

5. CONCLUSION

We use intent discovery approach for iUnit ranking subtask and the result is encouraging. Our iUnit summarizatoin method also considers intents. Our result submitted to the iUnit ranking subtask demonstrates a superior performance.

6. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[2] K. C.-T. Chang, Y.-H. Wu, Y.-L. Tsai, and R. T.-H. Tsai. Improving iunit retrieval with query classification and multi-aspect iunit scoring: The iisr system at ntcir-11 mobileclick task. In *Proceedings of NTCIR*, pages 208–212, 2014.

[3] J. He, E. Meij, and M. de Rijke. Result diversification based on query-specific cluster ranking. *Journal of the American Society for Information Science and Technology*, 62(3):550–571, 2011.

[4] 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 NTCIR-11*, pages 195–207, 2014.

[5] M. P. Kato, T. Sakai, T. Yamamoto, V. Pavlu, H. Morita, and S. Fujita. Overview of the ntcir-12 mobileclick task. In *Proceedings of NTCIR-12*, 2016.

[6] C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining (WSDM)*, pages 441–450. ACM, 2010.

[7] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.