IRIT at the NTCIR-12 MobileClick-2 Task

Abdelhamid. Chellal
IRIT Université de Toulouse III, France
abdelhamid.chellal@irit.fr

Mohand. Boughanem
IRIT Université de Toulouse III, France
mohand.boughanem@irit.fr

ABSTRACT

This paper presents the participation of IRIT laboratory (University of Toulouse) to MobileClick-2 Task of NTCIR-12. This task aims to provide immediate and direct information that can be accessed by users' mobiles. For a given query, summarization systems are expected to provide two-layered summary of relevant information units (iUnits). Two subtasks have been defined named as iUnit ranking and iUnit Summarization. In iUnit ranking subtask, we propose to rank iUnits according to their amount of information which are evaluated using Shannon's entropy. For iUnit summarization subtask, we propose two different strategies to build a summary. The first one is a top-down approach where the first layer is filled first, while the second strategy is bottom-up approach in which we start by filling the second layer. To estimate the similarity between words, we investigated the use of word2vec tool. For all these approaches, we discuss the obtained results during the experimental evaluation.

Team Name

IRIT

Subtasks

iUnit Ranking Subtask (English)
iUnit Summarization Subtask (English)

Keywords

iUnit Summarization, entropy, word2vec.

1. INTRODUCTION

In response to a query, users get usually a ranked list of URLs. Hence, to fulfill their information need, they have to explore many URLs to gather relevant pieces of information, since salient information are often located in several Web pages. However, exploring all returned URLs requires significant effort and attention, especially for mobile users. In such a scenario, it would be more beneficial, for the user, if information retrieval system returns a concise summary to the query that integrates several relevant fragments of information in an intelligible way.

The NTCIR-12 MobileClick task aims to facilitate the information access for mobile users. For a given query, systems are expected to provide two-layered summary of relevant information [2]. The first layer presents the most important information with outlining any additional relevant information. The second layers contain detailed information that can be accessed by clicking on an associated part of the text at the first layer.

The main purpose of generating a two-layered summary is to minimize the amount of text that the user has to read or, equivalently, minimize the time required to obtain the relevant information. To generate such summary, we have to overcome the following two issues:

- From a list of pieces of information (iUnit), how to select those which are worthy to be pushed in the first layer?
- For which intent (second layer) an iUnit will be assigned?

For these purposes, two subtasks were defined for MobileClick task, namely, iUnit ranking and summarization subtasks.

- The iUnit ranking subtask is a task where systems are expected to rank a set of information units (iUnits) based on their importance for a given query. The provided set of iUnits includes relevant as well as irrelevant iUnits which should be ranked below.
- The iUnit summarization subtask is defined as follows: Given a query, generate a structured textual output consisting on two layers. The first layer is a list of information units (iUnits) and links to the second layer, while the second layer consists of lists of iUnits. A link is one of the provided intents and it is associated with one of the iUnit lists in the second layers. Each list of iUnits in the first and second layers can include at most 420 characters (for English) so that it fits ordinary mobile screen size.

In this paper, we describe our participation to the NTCIR-12 MobileClick-2 task in which we submit several runs for both subtasks. However, our participation focuses only on the English language data set.

For the iUnits ranking subtasks, the proposed method ranks iUnits according to the amount of information carried by an iUnit. To estimate the amount of information of an iUnit, we use Shannon's entropy [5].

For iUnit summarization subtask, we propose two approaches to build two-layered summary according to the order in which the layers are filled. In the first approach, we adopt a top-down strategy which is similar to the baseline used by the organizers, while in the second proposed approach we utilize an bottom-up strategy. In the former strategy, we start by filling the first layer with the most important iUnits until the length limit is reached and the remaining iUnits are put in the second layer. In the later strategy, the second layer is filled first.

In both summarization approaches (Top-down and Bottom-up), the filling of the second layer is based on the similarity score between an intent and an iUnit. An iUnit is assigned to the most similar intent. We investigated two functions to estimate the similarity score between an intent and an iUnit. The first function is based on the word overlapping between iUnit and intent sets of words and the second function uses word2vec tool [3]. In each list of the first and second layers, iUnits are ranked according to the method described for the iUnits ranking subtasks.

The remainder of this paper is organized as follows. Section 2 introduces the proposed approaches for iUnit ranking subtask. Section 3 describes our iUnit two layer summarization strategies. Section 5 concludes the paper.

2. IUNIT RANKING

2.1 iUnit importance

In information theory, the amount of information carried by a message can be evaluated through Shannon's entropy [5]. To evaluate the importance of iUnit, we use the entropy measure by considering a document set on query. Our method ranks iUnits in order of their amount of information, which is assumed to present the retrieval status value (RSV) of an iUnit.

Let D_q be a document set on query q, the importance $RSV(u,q)$ of iUnit u regarding the query q is measured as follows:

$$RSV(u,q) = -\sum_{w \in u} P(w|D_q) \times log_2(P(w|D_q)) \quad (1)$$

Where $P(w|D_q)$ represents the probability of occurrence of term w in a document set on query q. This probability is estimated by Maximum likelihood estimation (MLE) as follows:

$$P(w|D_q) = \frac{fr_{D_q}(w)}{|D_q|} \quad (2)$$

Where $fr_{D_q}(w)$ is the frequency of word w in D_q, and $|D_q|$ is the number of words in D_q. We utilize noun in titles and summaries of search engine indexes to estimate this probability.

The intuition behind this proposition is that the iUnit is considered important (worthy to be presented in the top of a list) compared with other iUnits if the amount of information provided by this iUnit is greater than the amount of information supplied by others.

In order to boost the score of the words that occur in the query, we explore two variety of combinations of the entropy measure with local score of relevance of iUnits' word with respect to the query.

The first variety combines the probability of occurrence of the word w in the document set of query with its frequency of occurrence in the query as follows:

$$RSV(u,q) = -\sum_{w \in u} P(w|D_q) \times log_2(P(w|D_q)) \times (1 + fr_q(w)) \quad (3)$$

Where the $fr_q(w)$ is the number of occurrence of word w in the query q.

In the second variety, we combine the entropy score of word with its local relevance score with respect to the query. The local relevance score of a word w regrading the query q is estimated by computing its similarity score with all words that occur in the query q. We use word2vec tool [3] to estimate the similarity between two words as follows:

$$rsv(w_i, q) = \sum_{w_j \in q} word2vec\_similarity(w_i, w_j) \quad (4)$$

Hence, the global relevance score of the iUnit u regarding the query q is evaluated as follows:

$$RSV(u,q) = -\sum_{w \in u} P(w|D_q) \times log_2(P(w|D_q)) \times (1 + rsv(w,q)) \quad (5)$$

Word2vec is a tool that computes vector representations of words developed by Mikolov, Sutskever, Chen, Corrado and Dean in 2013 at Google Research. It is a two-layer neural network that processes text. Its input is a text corpus and its output is a set of vectors (feature vectors for words in that corpus) [3]. The purpose of Word2vec is to group the vectors of similar words together in vector space. The similarity between two words is measured by cosine similarity between their vector space. The idea is that words that share many contexts will be similar to each other.

In our participation, to train the word2vec model we use titles and summaries of search engine "indices" provided for all the query as input text to build the set of vectors.

2.2 Result for iUnit ranking

For iUnit ranking subtask, we submit several runs that are summarized in the table 1.

Table 1: Different run submitted for English iUnits ranking sub-task.

ID	File name	Ranking function
88	IRIT-rank-1	$-\sum_{w \in u} P(w\|D_q) \times log_2(P(w\|D_q)$
188	rank-entropy-v2	$-\sum_{w \in u} P(w\|D_q) \times log_2(P(w\|D_q)) \times (1 + fr_q(w))$
233	rank-entropy-v3	$\sum_{w \in u} P(w\|D_q) \times (1 + fr_q(w))$
341	IRIT-rank-w2v	$-\sum_{w \in u} P(w\|D_q) \times log_2(P(w\|D_q)) \times (1 + rsv(w,q))$

Table 2 reports the results of our different runs under two official metrics known as normalized discounted cumulative gain (nDCG) [1] and Q-measure proposed by Sakai [4]. Q-measure is a recall-based metric, while nDCG is a rank-based metric. nDCG is computed for 3, 5, 10 and 20 top iUnits.

First, we observe that all our runs outperform the organizer's baseline. It is clear that the use of the entropy measure provides better ranking than the baseline that used odd ratio between a document set on a query and the others. This results can be explained by the fact that the entropy measure gives high score for long iUnits that contain frequent words. Indeed, a long iUnit is more informative than a short one.

We notice also that taking into account the occurrence of the word in the query improves the ranking quality. The run 188 that combines the number of occurrence of word of iUnit in the query is the best performing overall under both metrics (nDCG and Q measure). This result was expected, because iUnits in which a query's words appear are often more preferable than the others that do not contain any query word. However, we see that the similarity score between the iUnit word and query word computed using word2vec tool dose not improve the ranking quality. It seems that the size

Table 2: Results for iUnits ranking. Rows are sorted by Q-measure .

ID	Run	nDCG@3	nDCG@5	nDCG@10	nDCG@20	Q
188	rank-entropy-v2	**0.7511**	**0.7698**	**0.8128**	**0.8773**	**0.9036**
88	IRIT-rank-1	0.7457	0.7665	0.8088	0.8757	0.9019
341	IRIT-rank-w2v-stem-5	0.7448	0.7667	0.809	0.8754	0.9018
233	rank-entropy-v3	0.7381	0.7577	0.8027	0.8735	0.8993
Baseline		**LM based ranking**				0.8975

of text corpus used to build the words vectors is not big enough to catch the similarity between words.

3. IUNIT SUMMARIZATION

According to the order in which the summary layers are filled, we investigate two strategies aiming at building two-layered summary using the entropy based method for iUnits ranking as described in previous section. The first strategy is a top-down approach where the first layer is filled first and puts lower-ranked iUnits in the second layer relevant to them. The second strategy is a bottom-up approach in which second layers are filled before the first layer.

3.1 Top-down summarization strategy

The iUnits are sorted according to the entropy method described in the previous section and the top-down approach puts the top-ranked iUnits in the first layer followed by all the intents which are used as links in the same order in the intent file. Lower-ranked iUnits (in order of the entropy) are put in the second layer relevant to them. The first and second layers are filled until the length of iUnits exceed the length limit (set to 420 characters for English task, with excluding symbols and white-spaces).

Each second layer corresponds to an intent. The remaining iUnits (not used in the first layer) are sorted in decreasing order of their score of relevance with an intent. For each intent, we put the top ranked iUnits in the related second layer until the total length exceeds the limit. Note that in this approach, the iUnit of the first layer does not appear in the second layer while the remaining iUnits may appear multiple times in the second layer, e.g. an iUnit may appear in two second elements related to two different intents.

Two different ways to estimate the score of relevance of iUnit with respect to an intent were investigated. The first one combines the entropy score of an iUnit with the asymmetric similarity between u and i as follows:

$$Score(u, i) = RSV(u, q) * Sim(u, i) \qquad (6)$$

where $RSV(u, q)$ is the retrieval status value of the iUnit u regarding the query q which can be any of the functions presented in the section (2.1), and $Sim(u, i)$ is asymmetric similarity between u and i which is estimated as follows:

$$Sim(u, i) = |W_u \cap W_i|/|W_i| \qquad (7)$$

where W_x is a set of words contained in x. To avoid giving 0 similarity score, a small value is returned if there is no overlap between the iUnit and intent.

The second relevance score function tested in our participation utilizes the similarity between words of the iUnits and the words of the an intent. We use word2vec tool to estimate the similarity between two words as follows:

$$Score(u, i) = \sum_{w_i \in u} \sum_{w_j \in i} word2vec\_similarity(w_i, w_j) \qquad (8)$$

The use of the query documents set as input to word2vec tool to build the words vector representation may yield to zero similarity score because words of intent may not appear in the query documents set. To overcome this issue, we use a general text corpus provided within word2vec tool as input to build words vectors for computing the similarity between words of the iUnits and words of an intent. This measure favors longer iUnits because we thing that longer one brings high amount of information than shorter iUnits and hence it should be presented in the top of list for user. Algorithm 1 describes the overview of our top-down summarization approach for a given query q.

Input:
Q : a query defined by a set of keyword
IU_list : a list of iUnits of the given query Q
I_list : a list of intents of the given query Q
$length\_limit = 420$ For English subtask
Output: $First\_layer$; $Second\_layer$
begin
    $First\_layer \leftarrow \emptyset$; $Second\_layer \leftarrow \emptyset$
    $IU\_score \leftarrow RSV(IU\_list, Q)$
    $Limit \leftarrow length\_limit - length(IU\_list)$
    $IU\_score \leftarrow Sort(IU\_score)$
    $First\_layer \leftarrow put\_unitl\_limit(IU\_score, Limit)$
    $First\_layer \leftarrow First\_layer + I\_list$
    $Remaining\_IU\_list \leftarrow IU\_list \setminus First\_layer$
    for $i \in I\_list$ **do**
        for $u \in Remaining\_IU\_list$ **do**
            $IU\_score\_list\_second[u] \leftarrow score(u, i)$
        end
        $Second\_layer[i] \leftarrow$
        $put\_unitl\_limit(IU\_score\_list\_second, limit)$
    end
end

Algorithm 1: Top-down summarization strategy.

3.2 Bottom-up summarizatrion strategy

In this approach, each intent corresponds to a link to second layer. This approach starts by assigning iUnits to each intent relevant to them. For a given intent, iUnits are sorted in decreasing order according to their similarity score with the intent. This similarity is computed using word2vec approach (equation 8). The top iUnits are added to the list of the second layer of the corresponding intent until the length of iUnits exceed the length limit. For the first layer, the iUnits are sorted according to their relevance score with respect to the query. The relevance score is estimated using the function described in the equation (5). The top iUnits is added, iteratively, to the first layer followed immediately by the intent which has the highest similarity score with the added iUnit (if this intent is not already added in the first layer). To avoid redundancy in the summary, the added iUnit in the first layer is removed form the list of the corresponding intent in the second layer.

Unlike the top-down approach where the same iUnit element may appear multiple times, in the bottom-up strategy an iUnit appears only one time. Algorithm 2 describes the bottom-up summarization approach for a given query q.

Input:
Q : a query defined by a set of keyword
IU_list : a list of iUnits of the given query Q
I_list : a list of intents of the given query Q
$length\_limit = 420$ For English subtask
Output: $First\_layer$; $Second\_layer$
begin
  $First\_layer \leftarrow \varnothing$; $Second\_layer \leftarrow \varnothing$
  for $i \in I\_list$ **do**
    $IU\_score\_second \leftarrow score(IU\_list, i)$
    $IU\_score\_second \leftarrow sort(IU\_score\_second)$
    $Second\_layer[i] \leftarrow$
    $put\_unitl\_limit(IU\_score\_second, length\_limit)$
  end
  $total\_length \leftarrow 0$
  $IU\_score \leftarrow RSV(IU\_list, Q)$
  $IU\_score\_list \leftarrow sort(IU\_score)$
  for $u \in IU\_score$ **do**
    $total\_length \leftarrow total\_length + length(u)$
    if $total\_length > length\_limit$ **then**
      $break$
    else
      $intent \leftarrow \underset{\forall i \in I\_list}{argmaxscore}(u, i)$
      $second\_layer[intent] \leftarrow$
      $second\_layer[intent] - \{u\}$
      $first\_layer \leftarrow first\_layer + \{u\}$
      if $intent \notin first_layer$ **then**
        $first\_layer \leftarrow first\_layer + intent$
    end
  end
end

Algorithm 2: Bottom-up summarization strategy.

Figure 1 shows an example of the first level of the summary generated for the query "MC2-E-0001" by both strategies (bottom-up and Top-down) in the left side and the right side respectively.

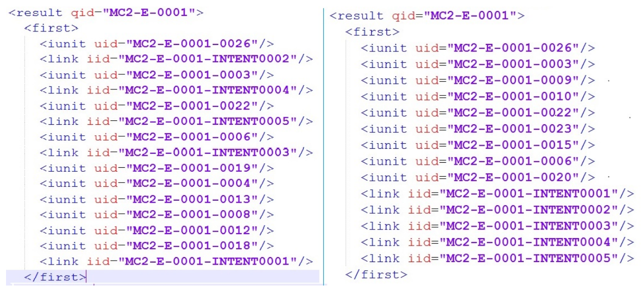
3.3 Result for iUnits summarization

Table 3 presents different configurations evaluated for iUnits summarization subtask. In columns 3, 4 and 5 we specify the layers filling strategy, the iUnit ranking and the similarity functions used respectively to generate the submitted runs.

Table 3: Configuration of different runs submitted for iUnits summarization.

ID	File name	strategy	Ranking	Similarity
400	IRIT-SUM-w2v-8	Top-down	Equation 5	word2vec
89	IRIT-sum-1.xml	Top-down	Equation 1	words overlap
358	IRIT-sum-w2v-7	Bottom-up	Equation 5	word2vec
230	IRIT-SUM-v2	Top-down	Equation 3	words overlap

Table 4 reports the performance of our runs which are compared with the performance of the organizer's baselines in term of the M measure. We observe that the best performance of our approach is achieved with a top-down strategy. The use of bottom-up strategy dose not improve the performance. This result can be explained by the fact that in top-down strategy intents (links to the second layer) are put at the end of the list while in the bottom-up strategy intents may be presented in the top of the first layer. In the later strategy, the first layer may be filled with low ranked iUnits because an iUnit that appears in the second level related to an intent will be excluded from the first layer. Also, we notice that the use of word2vec tool, instead of a word overlapping, to evaluate the similarity between an iUnit and an intent, improves the quality of the generated summary (run ID 400 vs run ID 89). However, our best performing run remains under the baseline based on language model.


```
<result qid="MC2-E-0001">
  <first>
    <iunit uid="MC2-E-0001-0026"/>
    <link iid="MC2-E-0001-INTENT0002"/>
    <iunit uid="MC2-E-0001-0003"/>
    <link iid="MC2-E-0001-INTENT0004"/>
    <iunit uid="MC2-E-0001-0022"/>
    <link iid="MC2-E-0001-INTENT0005"/>
    <iunit uid="MC2-E-0001-0006"/>
    <link iid="MC2-E-0001-INTENT0003"/>
    <iunit uid="MC2-E-0001-0019"/>
    <iunit uid="MC2-E-0001-0004"/>
    <iunit uid="MC2-E-0001-0013"/>
    <iunit uid="MC2-E-0001-0008"/>
    <iunit uid="MC2-E-0001-0012"/>
    <iunit uid="MC2-E-0001-0018"/>
    <link iid="MC2-E-0001-INTENT0001"/>
  </first>

<result qid="MC2-E-0001">
  <first>
    <iunit uid="MC2-E-0001-0026"/>
    <iunit uid="MC2-E-0001-0003"/>
    <iunit uid="MC2-E-0001-0009"/>
    <iunit uid="MC2-E-0001-0010"/>
    <iunit uid="MC2-E-0001-0022"/>
    <iunit uid="MC2-E-0001-0023"/>
    <iunit uid="MC2-E-0001-0015"/>
    <iunit uid="MC2-E-0001-0006"/>
    <iunit uid="MC2-E-0001-0020"/>
    <link iid="MC2-E-0001-INTENT0001"/>
    <link iid="MC2-E-0001-INTENT0002"/>
    <link iid="MC2-E-0001-INTENT0003"/>
    <link iid="MC2-E-0001-INTENT0004"/>
    <link iid="MC2-E-0001-INTENT0005"/>
  </first>
```

Figure 1: First level of the summary generated for the query "MC2-E-0001 by bottom-up strategy in the left side and top-down strategy in the right side

Table 4: Results for iUnits summarization. Rows are sorted by M-measure

ID	File name	summary strategy	M
400	IRIT-SUM-w2v-8	Top-down	16.8656
89	IRIT-sum-1.xml	Top-down	16.5628
358	IRIT-sum-w2v-7	Bottom-up	16.4654
230	IRIT-SUM-v2	Top-down	15.5659
LM Baseline		Top-down	**16.8975**
Randome Baseline		Top-down	14.1051

4. CONCLUSIONS

In this paper, we describe our approach to the NTCIR-12 MobileClick-2 task. We submit several runs for both iUnit ranking and summarization subtask. The proposed ranking method is based on the use of the entropy measure to evaluate the importance of an iUnit. For iUnit summarization, we propose two approaches according to the order in which the two layers of the summary are filled. The use of word2vec tool to estimate the relevance of an iUnit regarding an intent was investigated. For this first participation, the primarily results are promoting particularity for iUnit ranking. However, further study is needed to be carried out to enhance the quality of the generated summary, especially on how to assign an information fragment to a relevant intent.

5. REFERENCES

[1] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, Oct. 2002.

[2] T. Y. V. P. H. M. S. F. Makoto P. Kato, Tetsuya Sakai. Overview of the ntcir-12 mobileclick task. 2016.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[4] T. Sakai. On the reliability of information retrieval metrics based on graded relevance. *Inf. Process. Manage.*, 43(2):531–548, Mar. 2007.

[5] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 , 623–656, Oct. 1948.