# Overview of the NTCIR-11 MobileClick Task

Makoto P. Kato
Kyoto University
kato@dl.kuis.kyoto-u.ac.jp

Matthew Ekstrand-Abueg
Northeastern University
mattea@ccs.neu.edu

Virgil Pavlu
Northeastern University
vip@ccs.neu.edu

Tetsuya Sakai
Waseda University
tetsuyasakai@acm.org

Takehiro Yamamoto
Kyoto University
tyamamot@dl.kuis.kyoto-u.ac.jp

Mayu Iwata
KDDI Corporation
iwata.mayu@ist.osaka-u.ac.jp

## ABSTRACT

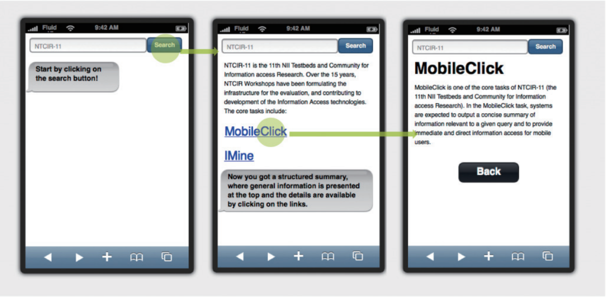
This is an overview of the NTCIR-11 MobileClick task (a sequel to 1CLICK in NTCIR-9 and NTCIR-10). In the MobileClick task, systems are expected to output a concise summary of information relevant to a given query and to provide immediate and direct information access for mobile users. We designed two types of MobileClick subtasks, namely, iUnit retrieval and summarization subtasks, in which four research teams participated and submitted 14 runs. We describe the subtasks, test collection, and evaluation methods and then report official results for NTCIR-11 MobileClick.

## 1. INTRODUCTION

Current web search engines usually return a ranked list of URLs in response to a query. After inputting a query and clicking on the search button, the user often has to visit several web pages and locate relevant parts within those pages. While these actions require significant effort and attention, especially for mobile users, they could be avoided if a system returned a concise summary of relevant information to the query [10].

The NTCIR-11 MobileClick task (and its predecessors, 1CLICK tasks organized in NTCIR-9 [11] and NTCIR-10 [4]) aims to directly return a summary of relevant information and immediately satisfy the user without requiring heavy interaction with the device. Unlike the 1CLICK tasks, we expect the output to be two-layered text where the first layer contains the most important information and an outline of additional relevant information, while the second layers contain detailed information that can be accessed by clicking on an associated anchor text in the first layer. As shown in Figure 1, for query "NTCIR-11", a MobileClick system presents general information about NTCIR-11 and a list of core tasks in the first layer. When the "MobileClick" link is clicked by the user, the system shows text in the second layer that is associated with that link.

Textual output of the MobileClick task is evaluated based on information units (iUnits) rather than document relevance. The performance of a submitted system is scored higher if it generates summaries including more important iUnits. In addition, we require systems to minimize the amount of text the user has to read or, equivalently, the time she has to spend in order to obtain relevant information. Although these evaluation principles were also taken into account in the 1CLICK tasks, here they are extended to two-layered summaries where users can read a summary in multiple ways. We assume a user model that reads different parts of the summary by probabilistically clicking on links and compute an evaluation metric based on the importance of iUnits read as well as the time spent to obtain them.



Start by clicking on the search button!

NTCIR-11 is the 11th NII Testbeds and Community for Information access Research. Over the 15 years, NTCIR Workshops have been formulating the infrastructure for the evaluation, and contributing to development of the Information Access technologies. The core tasks include:

MobileClick

IMine

Now you got a structured summary, where general information is presented at the top and the details are available by clicking on the links.

MobileClick

MobileClick is one of the core tasks of NTCIR-11 (the 11th NII Testbeds and Community for Information access Research). In the MobileClick task, systems are expected to output a concise summary of information relevant to a given query and to provide immediate and direct information access for mobile users.

**Figure 1: An application of the MobileClick task. Concise two-layered text can fit a small screen of the mobile device, and can be efficiently accessed by users of different interests.**

**Table 2: Important dates of MobileClick.**

| | |
|---|---|
| Jul 10, 2013 | Web page launch |
| Dec 11, 2013 | Sample queries and iUnits released |
| Mar 31, 2014 | Test queries released |
| May 31, 2014 | Run submissions due |
| Aug 31, 2014 | Evaluation results released |

MobileClick attracted four research teams from three countries: China, U.S.A., and Taiwan. Table 1 provides a list of NTCIR-11 MobileClick participants with the number of iUnit retrieval and summarization submissions. The total number of submissions was 14. Although we had Japanese subtasks as well and participants who registered to Japanese ones, no Japanese runs were submitted at this round.

Table 2 shows important dates of NTCIR-11 MobileClick. We first released sample queries and iUnits (important pieces of information for each query) to help potential participants better understand the MobileClick task. We then released test queries and a document collection from which participants are expected to extract iUnits and generate two-layered summaries. We received runs from participants by May 31 in 2014, and released evaluation results on August 31 in 2014.

The remainder of this paper is structured as follows. Section 2 describes the details of the iUnit retrieval and summarization subtasks. Section 3 introduces a test collection consisting of queries, iUnits, and a document collection. Section 5 describes our evaluation methodology. Section 6 reports on the official evaluation results for both subtasks. Finally, Section 7 concludes this paper.

**Table 1: NTCIR-11 MobileClick participants and the number of iUnit retrieval and summarization submissions.**

| Team name | iUnit retrieval | iUnit summarization | Organization |
|---|---|---|---|
| KPNM [12] | 3 | 0 | Hunan University of Science and Technology, China |
| IISR [3] | 1 | 0 | National Central University, Taiwan |
| udel [2] | 5 | 3 | University of Delaware, U.S.A. |
| NTOU [5] | 1 | 1 | National Taiwan Ocean University, Taiwan |
| Total | 10 | 4 | |

# 2. SUBTASKS

MobileClick comprises the iUnit retrieval and summarization subtasks. This section explains the two types of subtasks, and their input and output.

## 2.1 iUnit Retrieval Subtask

The iUnit retrieval subtask is a task where systems are expected to generate a ranked list of pieces of information (iUnits) based on their importance for a given query. This subtask was devised to enable *componentized* evaluation, where we could separately evaluate the performance of extracting important information pieces and summarizing them into two-layers.

There are two types of iUnit retrieval runs:

- MANDATORY Runs: Organizers provide a document collection for each query. Participants must generate a list of iUnits only from the collection. The importance of each iUnit can be estimated by any data resources.
- OPEN Runs (OPTIONAL): Participants may choose to search the live web on their own to generate a list of iUnits. Any run extracts iUnits from at least some privately-obtained web search results is considered as an OPEN run, even if it also uses the baseline data.

Although participants could submit OPEN runs as well, we required them to submit at least one MANDATORY run for comparing their systems in a reproducible setting. As a result, only MANDATORY runs were submitted at this round.

We provided a set of 50 queries and asked participants to submit, for each query, a list of extracted iUnits that are ordered by their estimated importance. More concretely, we accept a tab-delimited-values (TSV) file as an iUnit retrieval run, where each line must represents a single iUnit, and be of the following format:

$qid \quad iUnit \quad score \quad source$
$qid \quad iUnit \quad score \quad source$
....

where $qid$ is a query ID, $iUnit$ is the text content of a piece of information, $score$ is the importance of the iUnit, and $source$ is either a URL (for web documents used OPEN runs) or the document ID (for given documents used in MANDATORY runs) from which a participant generated the iUnit. In many ways, the iUnit retrieval runs are similar with TREC adhoc runs in that they are essentially a ranked list of the objects retrieved. The iUnits text objects are to be assessed for relevance by human annotators (section 4) and the runs evaluated using ranking measures (section 5)

## 2.2 iUnit Summarization Subtask

The iUnit summarization subtask is defined as follows: for a given query and a given list of iUnits ranked according to their importance, generate a structured textual output. In MobileClick, more precisely, the output must consist of two layers. The first layer includes text and links to second layers, while second layers just contain text. A link comprises anchor text and the ID of a second layer.

The output summary is expected to include more important information and to minimize the amount of text users have to read. For example,

- a summary that contains more important information earlier in the first layer is evaluated better;
- for a query with few subtopics, a summary that shows all the information in the first layer would get a higher score than one that separates information into text fragments in the second layer.
- for a query with many subtopics, a summary that hides the details of each subtopic in the second layer is evaluated better than one that shows all the information in the first layer, as users interested in different subtopics can save text they have to read.

There are two types of iUnit summarization runs like the iUnit retrieval subtask:

- MANDATORY Runs: Participants must use a iUnit list distributed by the organizers only to generate summaries. Note that any data resources can be used for estimating the importance of each iUnit.
- OPEN Runs (OPTIONAL): Participants may choose to search the live web on their own to generate summaries. Any run uses contents from at least some privately-obtained web search results is considered as an OPEN run, even if it also uses the baseline data.

Only MANDATORY runs were submitted also in the iUnit summarization subtask.

Participants were given a list of queries and a list of iUnits we provided, and were asked to generate a two-layered summary for each query. Each run must be a XML file that satisfies a DTD shown below:

```
<!ELEMENT results (sysdesc, result*)>
<!ELEMENT sysdesc (#PCDATA)>
<!ELEMENT result (firstlayer, secondlayer*)>
<!ELEMENT firstlayer (#PCDATA | link)*>
<!ELEMENT secondlayer (#PCDATA)>
<!ELEMENT link (#PCDATA)>
<!ATTLIST result qid ID #REQUIRED>
<!ATTLIST link id CDATA #REQUIRED>
<!ATTLIST secondlayer id CDATA #REQUIRED>
```

where

- The XML file includes a [results] node as the root node;
- The [results] node contains exactly one [sysdesc] node;
- The [results] node also contains [result] nodes, each of which corresponds a two-layered summary and has a [qid] attribute;
- A [result] node contains a [firstlayer] node and [secondlayer] nodes;
- The [firstlayer] node contains text and [link] nodes, which represents a link to a [secondlayer] node like "a" tag in HTML; and
- A [link] node has an attribute [id], which specifies a [secondlayer] to be linked. The [secondlayer] nodes has an attribute [id], and contains text.

An XML file example that satisfies the DTD is shown below:

```
<results>
  <sysdesc>
  Organizers' Baseline
  </sysdesc>
  <result qid="MC-E-0001">
  <firstlayer>
    Java...
    <link id="1">Classes</link>
    <link id="2">Just in Python</link>
  </firstlayer>
  <secondlayer id="1">
    static typing...
  </secondlayer>
  <secondlayer id="2">
    Python is difficult for beginners...
  </secondlayer>
  </result>
</results>
```

# 3. TEST COLLECTION

The NTCIR-11 MobileClick test collection includes queries, iUnits, and a document collection. We describe the details of those components in the following subsections.

## 3.1 Queries

The NTCIR-11 MobileClick test collection includes 50 English and 50 Japanese queries (see Appendix A for the complete lists). In order to make the task more interesting and to discourage simply returning the first paragraph of a Wikipedia entry for the given entity or the snippets returned by the search engine, many of the queries were designed to be highly specific, *e.g.* "java vs python text processing" and "cheap hotel manhattan july 4". This trial is based on one of the lessons learnt from the NTCIR-10 1CLICK-2 task: we observed that a simple baseline method using Wikipedia achieved high performance for simple named entity queries.

## 3.2 iUnits

Like the 1CLICK tasks held in the past NTCIR, we used iUnits as a unit of information in the MobileClick task. iUnits are defined as *relevant*, *atomic*, and *dependent* pieces of information, where

- *Relevant* means that an iUnit provides useful factual information to the user;
- *Atomic* means that an iUnit cannot be broken down into multiple iUnits without loss of the original semantics; and
- *Dependent* means that an iUnit can depend on other iUnits to be relevant.

Please refer to the 1CLICK-2 overview paper for the details of the definition [4].

Organizers manually extracted iUnits from a document collection that we explain in the next subsection. As this work requires careful assessment lasting for a long time and consideration on the three requirements of iUnits, we decided not to use crowdsourcing mainly due to low controllability and high education cost. For English queries, we hired three assessors for extracting iUnits by hand, who were trained well through assessment work on TREC Temporal Summarization Track [1]. For Japanese queries, Japanese organizers of this task extracted iUnits by ourselves. The total number of iUnits is 3,819 (76.4 iUnits per query) for English queries and 1,940 (38.8 iUnits per query) for Japanese queries.

The weight of each iUnit was also given by the assessors including the Japanese organizers. Each English iUnit was evaluated at a three-point scale (1-3) by an assessor who extracted it, while each Japanese iUnit was evaluated at a five-point scale (1-5) by all the four Japanese organizers. The *individual* weights of Japanese iUnits were summed up: thus, the weight of Japanese iUnits ranges from 4 to 20. Examples of iUnits for English queries are shown in Table 3.

In the iUnit retrieval subtask, participants were required to extract iUnits from a document collection, while in the iUnit summarization subtask, they were required to arrange iUnits in a two-layered summary. We released our extracted iUnits with their importance, and allowed participants to use them for the iUnit summarization subtask.

## 3.3 Documents

To provide participants with a document collection, we downloaded 500 top-ranked documents that were returned by Bing search engine in response to each query. This download was conducted from February 23, 2014 to March 7, 2014 (JST). The title, summary provided by the search engine, URL, and rank of documents were recorded and released along with the document collection. As we failed to access some of the documents, the number of downloaded documents per query is fewer than 500. The average number of documents for English queries is 366 and that for Japanese queries is 417.

# 4. ASSESSMENT METHODOLOGY

## 4.1 iUnit Extraction

We begin by extracting relevant facts from relevant documents within the initial document pool. These facts form our initial pool of gold standard iUnits (GiUnits), which are used as potential input for the summarization subtask as well as the primary pool for judgment of the ranking task.

In order to perform this extraction, assessors are shown each document from the corpus, stripped of all HTML using a links dump, alongside the list of all iUnits extracted so far. They may select text from the document to create new iUnits, edit the text before addition, and review and edit all previously created iUnits at any time. Additionally, they may form dependencies between iUnits as previously described. Finally, they assign an importance level to the iUnit as to the ability of the iUnits to answer the given query. An example of this interface can be seen in Figure 2. Once the assessor has finished extracting iUnits for a given document, then click

**Table 3: Examples of iUnits for NTCIR-11 MobileClick English queries. Query MC-E-0020 is "stevia safety".**

| Query ID | iUnit | Weight | Source |
|---|---|---|---|
| MC-E-0020 | There are some dangers and side effects when using stevia. | 3 | MC-E-0020-008.html |
| MC-E-0020 | refined stevia preparations allowed in food and drinks | 3 | MC-E-0020-001.html |
| MC-E-0020 | Stevia does interact with some other drugs. | 2 | MC-E-0020-009.html |
| MC-E-0020 | Stevia may have an anti-inflammatory effect. | 1 | MC-E-0020-011.html |
| MC-E-0020 | Stevia may help diarrhea. | 1 | MC-E-0020-011.html |

to proceed to the next document in descending rank order according to the corpus rankings.

## 4.2 iUnits Matching

This section explains methods of evaluating participant iUnits (PiUnits) as a function of their matching to gold standard iUnits (GiUnits). We hired three assessors and asked them to identify all GiUnits semantically covered by each PiUnit. We pooled top 400 PiUnits per run and merged them into a single list for each query. A PiUnit list for each query was assessed by a single assessor, and there is no redundant evaluation. We simultaneously identify the presence and position of GiUnits and refine our GiUnit set, adding ones that appear in submitted PiUnits but are not included in the existing nugget set. In the next section, we compute evaluation metrics based on these matches and relevance grades of the matched GiUnits. In order to achieve high accuracy of the matches between our vital strings and participant summaries, we used a manual matching system with an automatic prior simply to reduce the workload for the assessors.

Figure 3 shows an example of the entailment interface. The left side of the interface contains a list of PiUnits currently being evaluated, highlighted to show the current matches. The right side contains a scrollable and searchable list of all GiUnits for the query along with dependencies and judged importance.

In the PiUnit list, the blue highlighted text represents automatic matches. These are verified by the assessor during the assessment process. The green matches indicate manually assigned matches. For ease of correlation, hovering over a vital string highlights in yellow its current match in the summary, if one exists. As mentioned, new GiUnits can be added as well to ensure full coverage of the information in the PiUnits.

Although we performed matching on the original PiUnits regardless of the number matches to each PiUnit, in the evaluation measures we only count the first (by text offset) matching GiUnit.

## 4.3 Link Labeling and Relevance Judgment

Link labeling was conducted in order to estimate the click probability $P(l_j)$ used in our evaluation metrics. In this task, assessors were asked to label each anchor text of iUnit summarization runs at a three-point scale: irrelevant (0), partially relevant (1), and relevant (2). Two assessors were assigned to each link, and the click probability $P(l_j)$ was estimated as follows:

$$P(l_j) = \frac{1}{2m}\sum_{i=1}^{m} r_i(l_j), \qquad (1)$$

where $r_i(l_j)$ is a score of a link $l_j$ given by the $i$-th assessor, and $m$ is the number of assessors assigned to each label ($m = 2$ in our evaluation). A link is never clicked on in our user model if all the assessors labeled it as irrelevant.

GiUnit-link relevance judgment was conducted to determine the gain by GiUnits in the iUnit summarization evaluation. We asked two assessors to independently evaluate the relevance of GiUnits to anchor texts that link to second layers where the GiUnits first appear. GiUnits were considered as relevant to the anchor text if one of the assessors labeled them as relevant.

# 5. EVALUATION MEASURES

This section describes evaluation methodology used in the NTCIR-11 MobileClick tasks.

## 5.1 iUnit Retrieval Subtask

Runs submitted by participants include a ranked list of PiUnits for each query. We first identify GiUnits covered by each PiUnit. Although multiple GiUnits can be covered by a single PiUnit, we only evaluate the first GiUnit that appears in each PiUnit as the iUnit retrieval subtask aims to find iUnits, which must satisfy the condition of *atomicity*. From the manual matching assessment, we obtain a ranked list of GiUnits for a given list of PiUnits. More precisely, given a list of PiUnits for a query ($P = (p_1, p_2, \ldots)$), we have a list of GiUnits $G = (g_1, g_2, \ldots)$. Note that $g_i$ can be empty ($\epsilon$) since a PiUnit may not contain any GiUnits.

We devise an evaluation metric for the iUnit retrieval subtask based on the following two principles: 1) a run receives a higher score if the run ranks more important GiUnits at higher ranks; 2) redundant GiUnits do not improve the score (as with other novelty-aware evaluation metrics).

Based on the two principles above, a generalized evaluation metric for the iUnit retrieval evaluation is defined as follows:

$$\sum_{i=1}^{k} \mathrm{gain}(g_i)\mathrm{decay}(i), \qquad (2)$$

where $\mathrm{gain}(g_i)$ is the gain by a GiUnit $g_i$, $\mathrm{decay}(i)$ is a decay function based on the rank of $p_i$, and $k$ is a cutoff parameter. Since the second principle requires the gain to be zero if $g_i$ equals to $g_j$ ($j < i$), $\mathrm{gain}(g_i)$ satisfies the condition below:

$$\mathrm{gain}(g_i) = 0 \quad ((\exists j < i) g_i = g_j). \qquad (3)$$

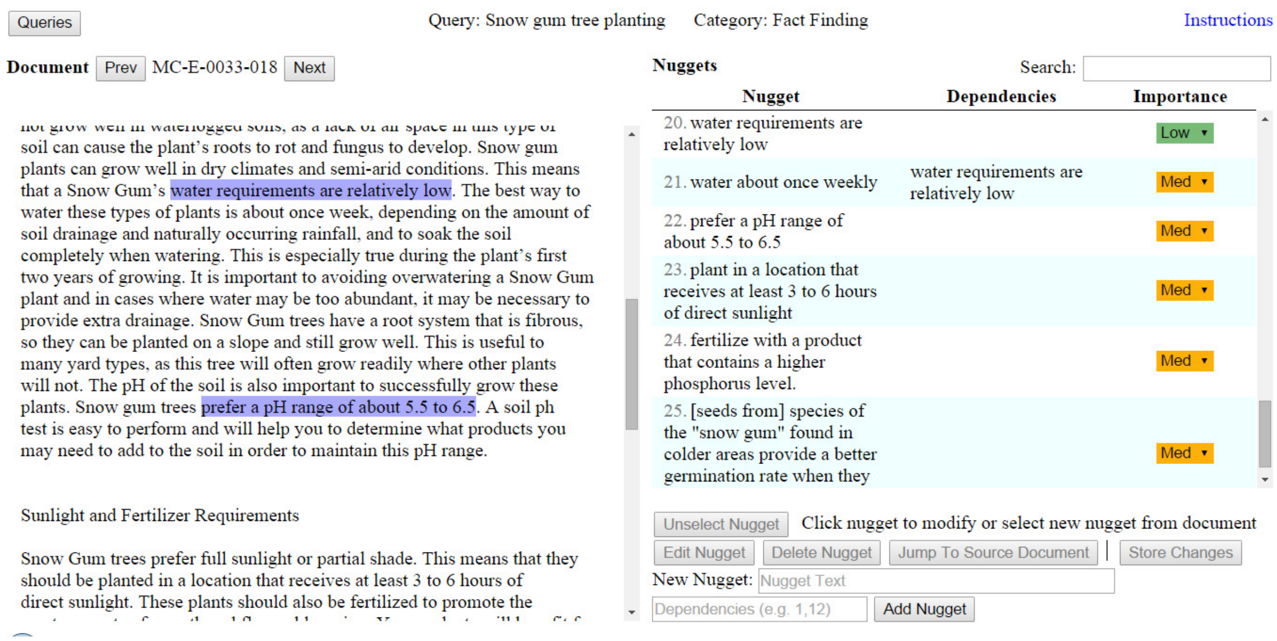
One of the implementations of Equation 2 is DCG:

$$\mathrm{DCG}@k = \sum_{i=1}^{k} \mathrm{gain}_w(g_i)\log_2(i+1)^{-1}, \qquad (4)$$

where $\mathrm{gain}_w(g_i) = w(g_i)$ $((\forall j < i) g_i \neq g_j)$; otherwise 0. Recall that $w(g_i)$ is the weight of a GiUnit $g_i$, and 0 if $g_i = \epsilon$. We use the standard normalized version of DCG (*i.e.* nDCG) as the primary evaluation metric in the iUnit retrieval subtask:

$$\mathrm{nDCG}@k = \frac{\mathrm{DCG}@k}{\sum_{i=1}^{k} \mathrm{gain}_w(g_i^*)/\log_2(i+1)}, \qquad (5)$$

where $g_j^*$ is the $j$-th GiUnit in an ideal ranked list. The ideal ranked list can be constructed by sorting all the GiUnits for a query by their weight.



Query: Snow gum tree planting Category: Fact Finding

Document MC-E-0033-018

not grow well in waterlogged soils, as a lack of air space in this type of soil can cause the plant's roots to rot and fungus to develop. Snow gum plants can grow well in dry climates and semi-arid conditions. This means that a Snow Gum's water requirements are relatively low. The best way to water these types of plants is about once week, depending on the amount of soil drainage and naturally occurring rainfall, and to soak the soil completely when watering. This is especially true during the plant's first two years of growing. It is important to avoiding overwatering a Snow Gum plant and in cases where water may be too abundant, it may be necessary to provide extra drainage. Snow Gum trees have a root system that is fibrous, so they can be planted on a slope and still grow well. This is useful to many yard types, as this tree will often grow readily where other plants will not. The pH of the soil is also important to successfully grow these plants. Snow gum trees prefer a pH range of about 5.5 to 6.5. A soil ph test is easy to perform and will help you to determine what products you may need to add to the soil in order to maintain this pH range.

Sunlight and Fertilizer Requirements

Snow Gum trees prefer full sunlight or partial shade. This means that they should be planted in a location that receives at least 3 to 6 hours of direct sunlight. These plants should also be fertilized to promote the

Nuggets

| Nugget | Dependencies | Importance |
|---|---|---|
| 20. water requirements are relatively low | | Low ▾ |
| 21. water about once weekly | water requirements are relatively low | Med ▾ |
| 22. prefer a pH range of about 5.5 to 6.5 | | Med ▾ |
| 23. plant in a location that receives at least 3 to 6 hours of direct sunlight | | Med ▾ |
| 24. fertilize with a product that contains a higher phosphorus level. | | Med ▾ |
| 25. [seeds from] species of the "snow gum" found in colder areas provide a better germination rate when they | | Med ▾ |

Click nugget to modify or select new nugget from document

New Nugget: Nugget Text

Dependencies (e.g. 1,12)

**Figure 2: MCEVAL system used for iUnit extraction.**

Another implementation of Equation 2 is Q-measure proposed by Sakai [7]:

$$Q = \sum_{i=1}^{|P|} \mathrm{gain}_Q(g_i)\mathrm{decay}_Q(i), \tag{6}$$

where

$$\mathrm{gain}_Q(g_i) = \begin{cases} \mathrm{rel}(i) + \beta \sum_{j=1}^{i} \mathrm{gain}_w(g_j) & (\mathrm{gain}_w(g_j) > 0), \\ 0 & (\mathrm{otherwise}), \end{cases} \tag{7}$$

(8)

$$\mathrm{decay}_Q(i) = \left( R \left( i + \beta \sum_{j=1}^{i} \mathrm{gain}_w(g_j^*) \right) \right)^{-1}. \tag{9}$$

In these equations, $\mathrm{rel}(i)$ is the number of GiUnits found within ranks $[1, i]$, $R$ is the total number of GiUnits, and $\beta$ is a patience parameter which we set to 1 following established standards [6].

Q-measure is a recall-based graded-relevance metric, while nDCG is a rank-based graded-relevance metric. Thus, we expect that using both metrics will enable us to measure the performance from different perspectives. Moreover, both of them were shown to be reliable [7].

## 5.2 iUnit Summarization Subtask

Runs submitted to the iUnit summarization subtask consists of the first layer $f$ and second layers $S = \{s_1, s_2, \ldots, s_n\}$. The first layer contains links $\mathbf{l} = (l_1, l_2, \ldots, l_n)$, which are sorted by their appearance position in the first layer. Each link $l_i$ has anchor text and links to a second layer $s_i$.

The principles of the iUnit summarization evaluation metric are summarized as follows:

(1) The evaluation metric is the expected utility of users who probabilistically read a summary.

(2) Users probabilistically read a summary following the rules below:

(a) They read the summary from the beginning of the first layer in order and stop after reading $L$ characters except symbols and white spaces.

(b) When they reach the end of a link $l_j$, they click on the link with a probability $P(l_j)$ and start to read the second layer $s_j$.

(c) When they reach the end of a second layer $s_j$, they continue to read the first layer from the end of the link $l_j$.

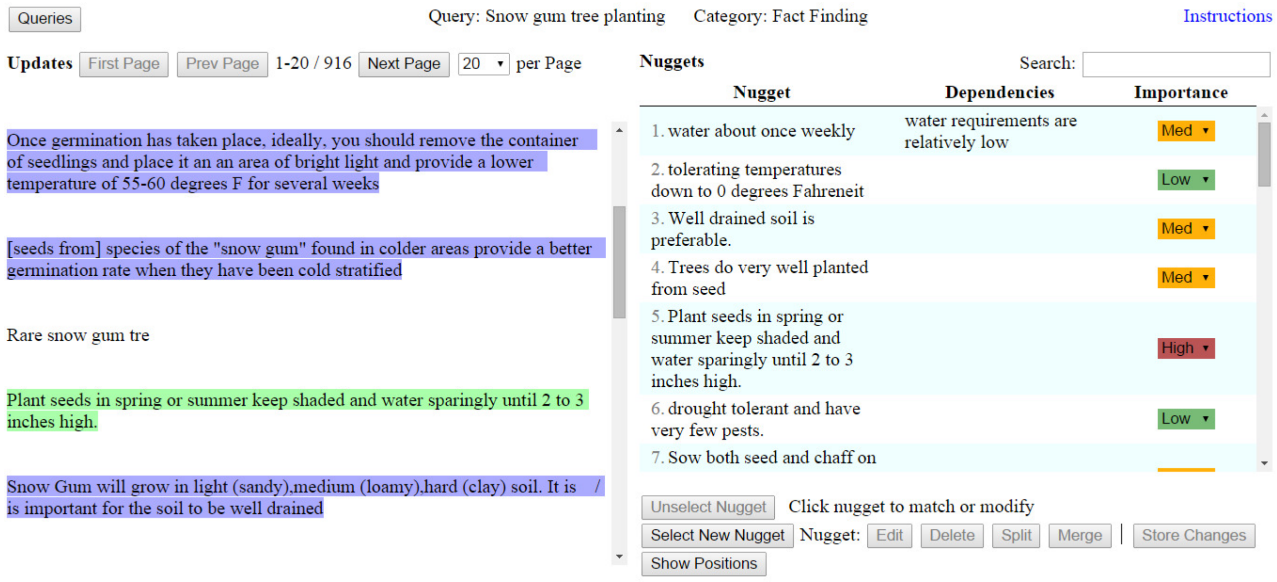
(3) The utility is measured by U-measure proposed by Sakai and Dou [9], which consists of a position-based gain and a position-based decay function.

We then generate the user tails (or *trailtext*) according to the user model explained above, compute a U-measure score for each trailtext, and finally estimate the expected U-measure by combining all the U-measure scores of different trailtexts. *M-measure*, the iUnit summarization evaluation metric, is defined as follows:

$$M = \sum_{t \in T} P(t)U(t), \tag{10}$$

where $T$ is a set of all possible trailtexts, $P(t)$ is a probability of going through a trail $t$, and $U(t)$ is the U-measure score of the trail.

A trailtext is a concatenation of all the texts read by the user. For the first layer $f$ including $n$ links and second layers, there are $2^n$ trailtexts as each link can be either clicked or not clicked on. For example, users can read this summary by 1) reading the first layer by the end of a link $l_j$, the second layer $s_j$, and the first layer from the end of the link $l_j$ to the end of the first layer; 2) reading the first layer by the end of a link $l_j$, the second layer $s_j$, the first layer from the end of the link $l_j$ to the end of a link $l_{j+1}$, the second layer $s_{j+1}$, and the first layer from the end of the link $l_{j+1}$ to the end of the first layer; and 3) reading only the first layer. Although it seems infeasible to use all the trailtexts, we can omit most of the trailtexts based on an assumption that users stop reading a summary after reading $L$ characters. In the following discussions, we use a *click*



Once germination has taken place, ideally, you should remove the container of seedlings and place it an an area of bright light and provide a lower temperature of 55-60 degrees F for several weeks

[seeds from] species of the "snow gum" found in colder areas provide a better germination rate when they have been cold stratified

Rare snow gum tre

Plant seeds in spring or summer keep shaded and water sparingly until 2 to 3 inches high.

Snow Gum will grow in light (sandy),medium (loamy),hard (clay) soil. It is / is important for the soil to be well drained

| Nugget | Dependencies | Importance |
|---|---|---|
| 1. water about once weekly | water requirements are relatively low | Med |
| 2. tolerating temperatures down to 0 degrees Fahreneit | | Low |
| 3. Well drained soil is preferable. | | Med |
| 4. Trees do very well planted from seed | | Med |
| 5. Plant seeds in spring or summer keep shaded and water sparingly until 2 to 3 inches high. | | High |
| 6. drought tolerant and have very few pests. | | Low |
| 7. Sow both seed and chaff on | | |

**Figure 3: Example Entailment Interface for query *"marvin gaye influence"* in English subtasks.**

*stream* to represent a trailtext $t$, which is defined as a list of binary values for links $\mathbf{l} = (l_1, l_2, \ldots, l_n)$:

$$\mathbf{c}^t = (c_1^t, c_2^t, \ldots, c_n^t). \quad (11)$$

The $j$-th value of $\mathbf{c}^t$ indicates whether the user clicks on a link $l_j$, and there is a one-to-one mapping between click streams and trailtexts. For example, a summary includes links $l_1$ and $l_2$ in this order. All the trailtexts are represented by click streams (0, 0), (1, 0), (0, 1), and (1, 1). The click stream (0, 1) indicates a trailtext where the user does not click on the link $l_1$, but click on the link $l_2$.

In our user model, users probabilistically click on links. Thus, the probability of reading a trailtext $t$ can be computed by the probability of clicking on each link. Letting $P(l_j)$ be the probability of clicking on a link $l_j$, for the summary including links $\mathbf{l} = (l_1, l_2, \ldots, l_n)$ in this order, the probability of a trailtext $t$ is defined as follows:

$$P(t) = \prod_{j=1}^{n} P(l_j)^{c_j^t}(1 - P(l_j))^{(1-c_j^t)}. \quad (12)$$

For example, a summary includes links $l_1$ and $l_2$ in this order, where $P(l_1) = 0.25$ and $P(l_2) = 0.5$. The probability of a trailtext represented by a click stream (0, 1) is $P(t) = (1 - 0.25) \times 0.5 = 0.375$.

The utility is measured by U-measure proposed by Sakai and Dou [9], and is computed by the weight and offset of GiUnits in a trailtext. In a similar way to the iUnit retrieval evaluation, we first identified GiUnits in each trailtext, and obtained a set of GiUnits $G_t$. Note that we did not extract any GiUnit from anchor text of the links, and texts after $L$ characters. The offset of a GiUnit $g$ in a trailtext is defined as the number of characters except symbols and white spaces between the beginning of the trailtext and the end of $g$ that first appear in the trailtext. According to Sakai and Dou's work [9], U-measure is defined as follows:

$$U = \frac{1}{\mathcal{N}} \sum_{g \in G_t} \mathrm{gain}_M(g) d_t(g), \quad (13)$$

where $d_t$ is a position-based decay function, and $\mathcal{N}$ is a normalization factor (which we simply set to 1). The position-based decay function is defined as follows:

$$d_t(g) = \max\left(0, 1 - \frac{\mathrm{pos}_t(g)}{L}\right), \quad (14)$$

where $\mathrm{pos}_t(g)$ is the offset of a GiUnit $g$ in a trailtext $t$. The gain $\mathrm{gain}_M(g)$ is basically defined as the weight of the GiUnit $g$, but is degraded if the GiUnit $g$ appears at a second layer, and is not relevant to the anchor text that links to the second layer. This is one of the unique points in the MobileClick task evaluation, and can be justified because users who click on a link and read a second layer behind the link would find GiUnits irrelevant if they are irrelevant to the anchor text of the link. Thus, $\mathrm{gain}_M(g)$ is defined as follows:

$$\mathrm{gain}_M(g) = \begin{cases} w(g) & (g \text{ first appears in the first layer,} \\ & \vee g \text{ is relevant to the anchor text } a(g)) \\ 0 & (\text{otherwise}), \end{cases} \quad (15)$$

where $a(g)$ indicates an anchor text that links to the second layer where $g$ first appears. The relevance of the GiUnit to the anchor text is manually judged as explained later.

## 5.3 Handling Dependency

As we explained in Section 3, iUnits can depend on other iUnits to be relevant. For example, iUnit "Famous Cathedral in Paris dating to the 13th century" highly depends on iUnit "Notre Dame". In other words, the former iUnit is relevant only if the latter iUnit appears in a list or a summary. Thus, we degraded the weight of iUnits in both of the subtasks: the weight of an iUnit was set to 0 if all the iUnits on which the iUnit depends do not appear in a list (iUnit retrieval), or in a trailtext (iUnit summarization). Although a primary method to handle the dependency between iUnits was the one we explained here, we also used some variants of the dependency handling method, *e.g.* ignoring all the dependencies.

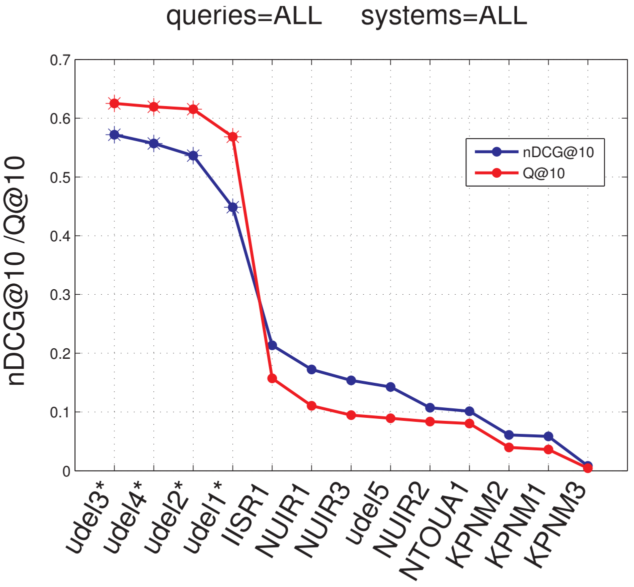


**Figure 4: system performance averaged over all queries. Runs marked with '*' are implementing a ranking function on the organizer-provided iUnits.**

# 6. RESULTS

## 6.1 Results for iUnit Retrieval Subtask

Retrieval evaluation is performed for all queries and runs and the evaluation metrics, nDCG and Q measure, are computed for a variety of cutoff thresholds $k$. The plot for nDCG, the primary measure, at $k = 10$ can be seen in Figure 4. The NUIR systems are baselines for comparison of performance to a set of standardized systems.

The baseline runs were constructed using three fairly naïve techniques, although they attempted to utilize text likely to be relevant. The first run, NUIR1, performed sentence segmentation on the snippets provided by the search results in the corpus. These sentences, in order of appearance in the results file, were used as the ranked list of iUnits. The second and third baselines looked for the first Wikipedia document in the search results, searching by URL, then the document was cleaned to plain text. The second baseline, NUIR2, then performed sentence segmentation on the resulting plain text Wikipedia document, and ranked the sentences in order of appearance in the document. The final baseline, NUIR3, ranked each sentence by the term frequency of query terms and returned the resulting list. If no Wikipedia document was found, no iUnits were returned in the ranked list. These baselines were intended to be relatively strong, but naïve solutions to the task. The results show that the baselines were strong, but there were systems which outperformed them.

Although udel performed best on average for four of its five runs, it is important to note that those runs can not be directly compared with the rest of the runs as the runs utilized the gold standard iUnits in their ranking methodology. The runs with an asterisk represent the runs which re-ranked the organizer-provided iUnits. The last run of that team did not use the gold iUnits and the performance is on par with the other systems and the baselines. Otherwise the methodology for their last run is similar to that of the other runs; see the participant paper for more information. IISR outperformed the baselines, but not by a significant margin.

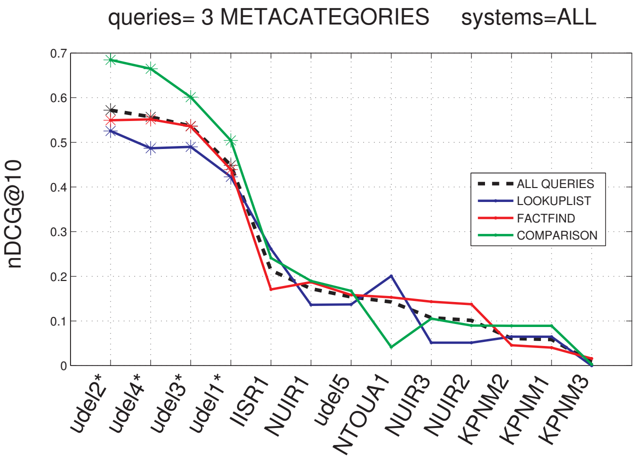


**Figure 5: system performance broken per query metacategories. Runs marked with '*' are implementing a ranking function on the organizer-provided iUnits.**

Additionally, the queries can be broken down into 4 main categories by the style of answer expected, as stated in the corpus description. It is clear from Figure 5 that participants performed similarly on the various query classes, but that there may be some inherent differences between the categories. For instance, the udel runs based on gold iUnits seem to perform better for comparison queries. Perhaps the ranking based on similarity of an individual iUnit to the global set of iUnits is easier for comparison queries because comparison queries contain more repeated words across iUnits, e.g. the name of the two items being compared, than lookup queries, which may only contain disjoin answers to a query object.

## 6.2 Results for iUnit Summarization Subtask

Table 4 shows submitted runs and descriptions of developed systems that were written in *sysdesc* node. The organizers provided a baseline based on the HTML structure of the distributed document collection. A basic strategy of the baseline is to output iUnits in descending order of the iUnit weight. Headers of an webpages in the collection were used as anchor texts. There are six levels of HTML headers: h1, h2, h3, h4, h5, and h6. We used the highest level that satisfies the following conditions: 1) the length of headers should be shorter than 40, and 2) the number of headers should be two or more. Our baseline first puts the headers in the first layer, and iUnits in descending order of the iUnit weight before the headers. As the length of each layer was limited to 280 characters, we stop outputting iUnits in the first layer when the length reaches the limit. Each anchor text has a second layer, where we output iUnits similar to the texts that follows the headers in webpages. More precisely, we first compute the similarity between each iUnit and a used header plus text that follows the header in a webpage. The similarity is defined as follows: $|X \cap Y|/|X|$, where $X$ is a set of words in an iUnit and $Y$ is a set of words in a header plus following text. We then output unused iUnits in descending order of their similarity to the length limit.

Figure 6 shows $M$ of each run with different values for $L$. Runs are sorted in descending order of $M$. For all the values for $L$, the order of runs is the same: SUM-ORG-E-MAND-1, SUM-NTOUA-E-MAND-1, SUM-udel-E-MAND-4, SUM-udel-E-MAND-1, and SUM-udel-E-MAND-5. Randomized Tukey's HSD test [8] shows that there are significant differences between all the pairs except ones between udel's runs.

**Table 4: Submitted runs and descriptions of developed systems.**

| Run | Description |
|---|---|
| SUM-NTOUA-E-MAND-1 | Grouping by longest leading substring. |
| SUM-udel-E-MAND-1 | Simple re-ranking approach based on the cosine similarity between each iUnit and a dynamic 'model' pseudo-document; At each step, Model doc is built using concatenation of iUnits that have been ranked so far, then select the doc least similar to model doc. |
| SUM-udel-E-MAND-4 | Simple re-ranking approach based on the cosine similarity between each iUnit and a fixed 'model' pseudo-document;model doc is constructed using the concatenation of top-10 docs for the query. |
| SUM-udel-E-MAND-5 | Simple re-ranking approach based on the cosine similarity between each iUnit and a fixed 'model' pseudo-document; Model doc is built using all concatenated iUnits. These iUnits are constructed by ourselves by consecutive tokens from top-10 docs with a max of 70 characters long. |
| SUM-ORG-E-MAND-1 | Organizers' Baseline: This method outputs gold standard iUnits in descending order of iUnit scores in the first layer, uses headers that appear at the same level in a HTML, and outputs iUnits similar to the text that follows the headers in the second layers. |

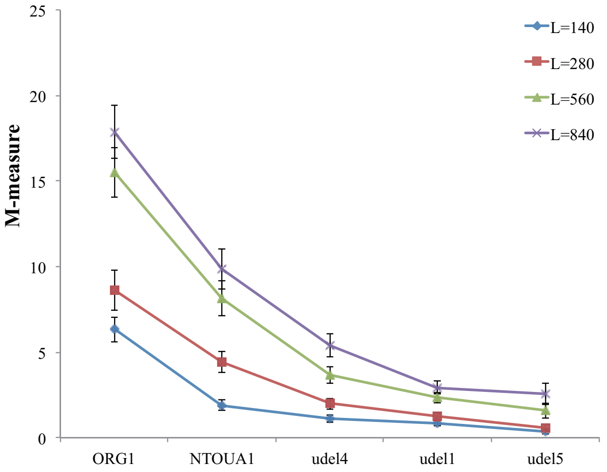


**Figure 6:** $M$ **of each run with different values for** $L$ **(±SEM)**

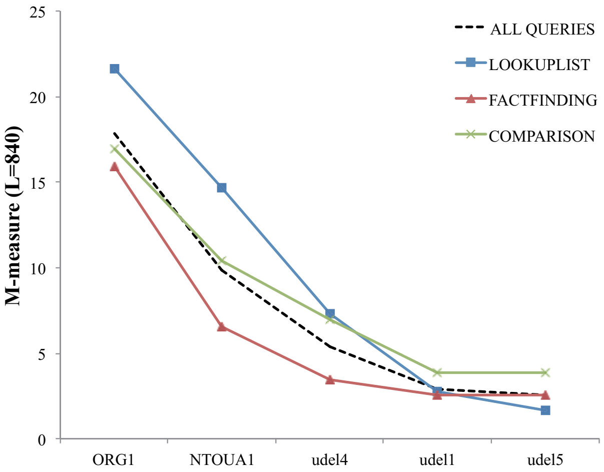


**Figure 7: Per-category** $M$ **(**$L = 860$**) of each run.**

Furthermore, we drilled down the results by the category of queries. Figure 7 shows per-category $M$ ($L = 860$) of each run. The trend of each category in the iUnit summarization subtask seems different from that in the iUnit retrieval subtask: MobileClick systems performed well for LOOKUPLIST, while they could not achieve high performances for FACTFINDING.

## 7. CONCLUSIONS

This paper presents the overview of the MobileClick task at NTCIR-11. This task aims to develop a system that returns a concise summary of information relevant to a given query, and brings a structure into the summarization so that users can easily locate their desired information. Our task attracted four teams and received fourteen runs for the iUnit retrieval and summarization subtasks. In this paper, we mainly explained the task design, evaluation methodology, and analysis of the results. We have a plan to continue the MobileClick task at NTCIR-12, and look forward to an improvement in the performance at the next round.

## 8. ACKNOWLEDGMENTS

We thank the NTCIR-11 MobileClick participants for their effort in submitting the runs, and NTCIR-11 PC chairs for their great organization including pre-task discussion and suggestions regarding the task organization.

## 9. REFERENCES

[1] J. Aslam, M. Ekstrand-Abueg, V. Pavlu, F. Diaz, and T. Sakai. TREC 2013 temporal summarization. In *Proceedings of the 22nd Text Retrieval Conference (TREC)*, 2013.

[2] A. Bah Rabiou and B. Carterette. Udel @ NTCIR-11 MobileClick Track. In *Proc. of NTCIR-11 Conference*, 2014.

[3] 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 *Proc. of NTCIR-11 Conference*, 2014.

[4] M. P. Kato, M. Ekstrand-Abueg, V. Pavlu, T. Sakai, T. Yamamoto, and M. Iwata. Overview of the NTCIR-10 1CLICK-2 Task. In *NTCIR-10 Conference*, pages 243–249, 2013.

[5] C.-T. Liu and C.-J. Lin. Description of the NTOU MobileClick System at NTCIR-11. In *Proc. of NTCIR-11 Conference*, 2014.

[6] T. Sakai. On penalising late arrival of relevant documents in

information retrieval evaluation with graded relevance. In *Proceedings of the First Workshop on Evaluating Information Access (EVIA 2007)*, pages 32–43, 2007.

[7] T. Sakai. On the reliability of information retrieval metrics based on graded relevance. *Information processing & management*, 43(2):531–548, 2007.

[8] T. Sakai. Statistical reform in information retrieval? *SIGIR Forum*, 48(1):3–12, 2014.

[9] T. Sakai and Z. Dou. Summaries, ranked retrieval and sessions: a unified framework for information access evaluation. In *Proc. of SIGIR 2013*, pages 473–482, 2013.

[10] T. Sakai, M. P. Kato, and Y.-I. Song. Click the search button and be happy: Evaluating direct and immediate information access. In *Proc. of CIKM 2011*, pages 621–630, 2011.

[11] T. Sakai, M. P. Kato, and Y.-I. Song. Overview of NTCIR-9 1CLICK. In *Proceedings of NTCIR-9*, pages 180–201, 2011.

[12] D. Zhou, Z. Wang, Z. Zeng, and T. Peng. KPNM at the NTCIR-11 MobileClick Task. In *Proc. of NTCIR-11 Conference*, 2014.

# APPENDIX

## A. QUERIES

Full lists of queries for English and Japanese MobileClick tasks are shown in Tables 5 and 6, respectively.

## B. RETRIEVAL EVALUATION RESULTS

Full lists of evaluation results for various $k$ values for nDCG and Q measure as well as number of ranked iUnits per run.

**Table 5: NTCIR-11 MobileClick English queries.**

| ID | Query |
|---|---|
| MC-E-0001 | java vs python text processing |
| MC-E-0002 | hiphop clubs barcelona |
| MC-E-0003 | why does turkey make you sleepy |
| MC-E-0004 | french landmarks |
| MC-E-0005 | Michoacan crafts materials |
| MC-E-0006 | ron paul tea party |
| MC-E-0007 | syrian civil war players |
| MC-E-0008 | best computer working position |
| MC-E-0009 | difference between junior college and community college |
| MC-E-0010 | sears illinois |
| MC-E-0011 | aaron rodgers belt celebration |
| MC-E-0012 | ukraine debt |
| MC-E-0013 | removing glue sticker |
| MC-E-0014 | growing seedless fruit |
| MC-E-0015 | best summer camping places in US |
| MC-E-0016 | home depot lowes hiring |
| MC-E-0017 | Snow gum tree planting |
| MC-E-0018 | boston bombing motive |
| MC-E-0019 | why vacuum insulates |
| MC-E-0020 | stevia safety |
| MC-E-0021 | why is apple developing maps |
| MC-E-0022 | mechanical keyboard benefits |
| MC-E-0023 | high protein pasta alternatives |
| MC-E-0024 | why do we yawn |
| MC-E-0025 | kofi annan syria |
| MC-E-0026 | led bulb 100 watt difficulty |
| MC-E-0027 | ivy bridge vs haswell |
| MC-E-0028 | power cord length limits |
| MC-E-0029 | government shutdown financial impact |
| MC-E-0030 | why UK does not adopt euro |
| MC-E-0031 | robert kennedy cuba |
| MC-E-0032 | healthy processed foods |
| MC-E-0033 | cheap hotel manhattan july 4 |
| MC-E-0034 | data mining course online |
| MC-E-0035 | bombay christian churches |
| MC-E-0036 | how is trash processed |
| MC-E-0037 | best art colleges connecticut |
| MC-E-0038 | book price fixing |
| MC-E-0039 | cheap home contractors miami |
| MC-E-0040 | marijuana legalization effects |
| MC-E-0041 | theaters texarkana |
| MC-E-0042 | marvin gaye influence |
| MC-E-0043 | concrete delivery nashua NH |
| MC-E-0044 | what is occupy wall street |
| MC-E-0045 | fedex hub TN |
| MC-E-0046 | ski resorts new england |
| MC-E-0047 | pope francis humility |
| MC-E-0048 | obamacare penalty |
| MC-E-0049 | russell crowe movies |
| MC-E-0050 | starbucks san francisco |

**Table 6: NTCIR-11 MobileClick Japanese queries.**

| ID | Query |
|---|---|
| MC-J-0001 | 世界で初めてノーベル賞をとった人は誰か |
| MC-J-0002 | 岡山駅カラオケ |
| MC-J-0003 | カーサ・ディ・ナポリ |
| MC-J-0004 | ハヤシライスの作り方 |
| MC-J-0005 | 未来科学館 |
| MC-J-0006 | なぜ空は青いのか |
| MC-J-0007 | 「くすぐったい」という感覚はどのようにして引き起こされるか |
| MC-J-0008 | 小池百合子キャスター |
| MC-J-0009 | 急がば回れ |
| MC-J-0010 | マイケルジャクソン死 |
| MC-J-0011 | 京都真如堂 |
| MC-J-0012 | なぜ猫はのどを鳴らすのか |
| MC-J-0013 | 地熱発電 |
| MC-J-0014 | 宮部みゆきドラマ |
| MC-J-0015 | 川口市交番 |
| MC-J-0016 | エコノミークラス症候群予防方法 |
| MC-J-0017 | 太宰治晩年 |
| MC-J-0018 | 小金井図書館 |
| MC-J-0019 | 宇都宮駅焼き鳥 |
| MC-J-0020 | 東淀川区眼科 |
| MC-J-0021 | 栗山千明カーネーション |
| MC-J-0022 | ザ・ペニンシュラ東京 |
| MC-J-0023 | ホテルアンビア松風閣 |
| MC-J-0024 | 横浜市役所 |
| MC-J-0025 | 京都市スーパー銭湯 |
| MC-J-0026 | 千葉いすみ市ペットショップ |
| MC-J-0027 | ダルビッシュ移籍 |
| MC-J-0028 | 鳥取王将 |
| MC-J-0029 | 一次遅れ |
| MC-J-0030 | ルソー社会契約論 |
| MC-J-0031 | 新垣結衣恋空 |
| MC-J-0032 | 福田歯科医院京都 |
| MC-J-0033 | 小林賢太郎うるう |
| MC-J-0034 | フォークリフトの免許の取り方 |
| MC-J-0035 | 羅生門効果 |
| MC-J-0036 | 牡蠣食べ放題横浜 |
| MC-J-0037 | トライオードアンプ |
| MC-J-0038 | 顔がむくむ病気 |
| MC-J-0039 | 微小生物アロメトリー |
| MC-J-0040 | ドラッガー経営理論 |
| MC-J-0041 | レンジでできるおかゆの作り方 |
| MC-J-0042 | ホテル西洋銀座歴史 |
| MC-J-0043 | 仁和寺仁王像 |
| MC-J-0044 | ワンレンボブ |
| MC-J-0045 | マールとフィーヌの違い |
| MC-J-0046 | PS2の止まる原因 |
| MC-J-0047 | ウルガモス |
| MC-J-0048 | 四日市ラーメン |
| MC-J-0049 | thoughとwhileの違い |
| MC-J-0050 | 妻夫木のび太役 |

**Table 7: Number of ranked iUnits by run.**

| TeamID | RunID | # Retrieved |
|---|---|---|
| IISR | 1 | 51.5800 (30.2325) |
| KPNM | 1 | 3599.2400 (2790.3009) |
| KPNM | 2 | 3590.6000 (2789.9506) |
| KPNM | 3 | 9.3200 (17.6867) |
| NTOUA | 1 | 27.9800 (10.9444) |
| NUIR | 1 | 20.0000 (0.0000) |
| NUIR | 2 | 16.8800 (6.5104) |
| NUIR | 3 | 17.5200 (5.7628) |
| udel | 1 | 76.3800 (63.9584) |
| udel | 2 | 76.3800 (63.9584) |
| udel | 3 | 76.3800 (63.9584) |
| udel | 4 | 76.3800 (63.9584) |
| udel | 5 | 295.3200 (32.7600) |

**Table 8: Mean (std) nDCG results for iUnit Retrieval Subtask.**

| TeamID | RunID | nDCG@5 | nDCG@10 | nDCG@80 | nDCG@400 |
|---|---|---|---|---|---|
| IISR | 1 | 0.2197 (0.2485) | 0.2134 (0.2197) | 0.1929 (0.1391) | 0.1809 (0.1336) |
| KPNM | 1 | 0.0647 (0.1562) | 0.0583 (0.1274) | 0.0747 (0.0928) | 0.1467 (0.1054) |
| KPNM | 2 | 0.0681 (0.1565) | 0.0609 (0.1276) | 0.0763 (0.0935) | 0.1480 (0.1059) |
| KPNM | 3 | 0.0068 (0.0276) | 0.0081 (0.0247) | 0.0058 (0.0154) | 0.0057 (0.0150) |
| NTOUA | 1 | 0.1529 (0.2087) | 0.1426 (0.1854) | 0.0927 (0.1124) | 0.0841 (0.1058) |
| NUIR | 1 | 0.1834 (0.1969) | 0.1723 (0.1740) | 0.1328 (0.1053) | 0.1224 (0.0947) |
| NUIR | 2 | 0.1083 (0.2041) | 0.1011 (0.1935) | 0.0694 (0.1115) | 0.0608 (0.0964) |
| NUIR | 3 | 0.1195 (0.1897) | 0.1073 (0.1609) | 0.0726 (0.1070) | 0.0655 (0.0983) |
| udel | 1 | 0.4399 (0.2437) | 0.4485 (0.2191) | 0.4578 (0.1691) | 0.4354 (0.1435) |
| udel | 2 | 0.5591 (0.2960) | 0.5720 (0.2647) | 0.6768 (0.2040) | 0.6915 (0.1718) |
| udel | 3 | 0.5200 (0.2864) | 0.5365 (0.2613) | 0.6645 (0.2066) | 0.6787 (0.1745) |
| udel | 4 | 0.5534 (0.3137) | 0.5570 (0.2855) | 0.6732 (0.2156) | 0.6906 (0.1811) |
| udel | 5 | 0.1602 (0.2113) | 0.1538 (0.1666) | 0.1679 (0.1261) | 0.2185 (0.1399) |

**Table 9: Mean (std) Q measure results for iUnit Retrieval Subtask.**

| TeamID | RunID | Q@5 | Q@10 | Q@80 | Q@400 |
|---|---|---|---|---|---|
| IISR | 1 | 0.1892 (0.2459) | 0.1573 (0.2041) | 0.0647 (0.0679) | 0.0546 (0.0585) |
| KPNM | 1 | 0.0520 (0.1215) | 0.0361 (0.0828) | 0.0149 (0.0281) | 0.0125 (0.0211) |
| KPNM | 2 | 0.0573 (0.1249) | 0.0396 (0.0857) | 0.0156 (0.0291) | 0.0131 (0.0216) |
| KPNM | 3 | 0.0067 (0.0260) | 0.0042 (0.0147) | 0.0008 (0.0027) | 0.0007 (0.0024) |
| NTOUA | 1 | 0.1063 (0.1517) | 0.0803 (0.1220) | 0.0222 (0.0329) | 0.0179 (0.0274) |
| NUIR | 1 | 0.1440 (0.1726) | 0.1105 (0.1377) | 0.0350 (0.0346) | 0.0293 (0.0278) |
| NUIR | 2 | 0.1019 (0.2030) | 0.0836 (0.1838) | 0.0229 (0.0473) | 0.0163 (0.0303) |
| NUIR | 3 | 0.1367 (0.2148) | 0.0946 (0.1548) | 0.0273 (0.0478) | 0.0222 (0.0419) |
| udel | 1 | 0.6264 (0.2776) | 0.5684 (0.2712) | 0.3455 (0.1914) | 0.3028 (0.1617) |
| udel | 2 | 0.6297 (0.2944) | 0.6153 (0.2824) | 0.5660 (0.2706) | 0.5503 (0.2599) |
| udel | 3 | 0.6450 (0.3053) | 0.6253 (0.3025) | 0.5684 (0.2742) | 0.5493 (0.2651) |
| udel | 4 | 0.6484 (0.3171) | 0.6195 (0.3114) | 0.5580 (0.2818) | 0.5450 (0.2698) |
| udel | 5 | 0.1164 (0.1941) | 0.0893 (0.1345) | 0.0370 (0.0487) | 0.0324 (0.0399) |

## C. SUMMARIZATION EVALUATION RESULTS

The average $M$ and standard deviation are shown in Tables 10 and 11. Table 12 shows optional metrics in the iUnit summarization subtask.

**Table 10: $M$ of each run with different values for $L$ (SD)**

| | Patience parameter | | | |
|---|---|---|---|---|
| | L=140 | L=280 | L=560 | L=840 |
| ORG1 | 6.33 (5.17) | 8.59 (8.20) | 15.4 (10.1) | 17.8 (10.9) |
| NTOUA1 | 1.91 (2.32) | 4.43 (4.36) | 8.14 (7.18) | 9.84 (8.24) |
| udel4 | 1.10 (1.60) | 1.98 (2.28) | 3.65 (3.44) | 5.39 (4.68) |
| udel1 | 0.83 (1.09) | 1.27 (1.23) | 2.34 (2.12) | 2.92 (2.62) |
| udel5 | 0.34 (1.01) | 0.57 (1.39) | 1.57 (3.04) | 2.55 (4.34) |

**Table 11: Per-category $M$ ($L = 860$) of each run (SD).**

| | Query category | | | |
|---|---|---|---|---|
| | ALL QUERIES | LOOKUPLIST | FACTFINDING | COMPARISON |
| ORG1 | 17.8 (5.17) | 21.6 (7.35) | 15.8 (11.0) | 16.9 (10.1) |
| NTOUA1 | 9.84 (2.32) | 14.7 (4.83) | 6.55 (4.13) | 10.3 (9.87) |
| udel4 | 5.39 (1.60) | 7.32 (2.88) | 3.45 (1.54) | 6.99 (7.02) |
| udel1 | 2.92 (1.09) | 2.78 (0.90) | 2.57 (1.21) | 3.85 (4.01) |
| udel5 | 2.55 (1.01) | 1.63 (0.70) | 2.53 (2.36) | 3.86 (6.60) |

**Table 12: Optional metrics in the iUnit summarization subtask (SD).**

| | Metrics | | | | |
|---|---|---|---|---|---|
| | # of iUnits | Sum of iUnit weights | $U$-measure of only the first layer | # of second layers | Average click probability |
| ORG1 | 44.9 (34.5) | 73.2 (48.4) | 10.5 (7.63) | 4.36 (3.44) | 0.19 (0.17) |
| NTOUA1 | 16.7 (14.9) | 25.2 (15.9) | 4.43 (4.41) | 1.62 (0.77) | 0.54 (0.38) |
| udel4 | 67.6 (62.6) | 96.3 (61.6) | 1.14 (1.34) | 12.7 (8.11) | 0.28 (0.12) |
| udel1 | 71.1 (58.6) | 95.0 (53.7) | 0.67 (1.03) | 8.92 (5.05) | 0.16 (0.12) |
| udel5 | 17.0 (20.4) | 32.9 (35.8) | 0.26 (0.72) | 61.0 (7.99) | 0.39 (0.29) |