# TTOKU Summarization Based Systems at NTCIR-9 1CLICK task

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

Takuya Makino
Tokyo Institute of Technology
Kanagawa, Japan
makino@lr.pi.titech.ac.jp

Tetsuya Sakai
Microsoft Research Asia
Beijing, China
tetsuyasakai@acm.org

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

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

## ABSTRACT

We describe our two query-oriented summarization systems implemented for the NTCIR-9 1CLICK task. We regard a Question Answering problem as a summarization process. Both of the systems are based on the integer linear programming technique, and consist of an abstractive summarization model and a model ensuring to cover diversified aspects for answering user's complex question. The first system adopts QSBP (Query SnowBall with word pair) for query oriented summarization, and extends the method to abstractive summarization in order to recognize and extract the parts of a sentence that are related to the query. On the other hand, The second system ensures covering pre-defined several aspects of information needed by users. We decided the types of aspects depending on the category of a query. Our first and second systems achieved 0.1585 and 0.1484 S-measure(I) score respectively in the desktop task. Furthermore, our first and second systems achieved 0.0866 and 0.0829 S-measure(I) score respectively in the mobile task.

## Keywords

Summarization, Integer Linear Programming

## 1. INTRODUCTION

Automatic text summarization has been studied for decades. Although newspaper articles have been chosen as the target text of summarization studies, web text becomes increasingly important as the target text, because web texts are increasing at an explosive pace. First, we propose a method that discriminate where the query relates and extract the necessary parts of a sentence. Web texts tend to have redundant or irrelative parts with important description. Particularly, we should remove unnecessary parts that spoil coherence or readability. Abstractive summarization also becomes important when we generate a summary from web texts.

We also assumed that users want to know various facts concerning their queries. That is to say, a generated summary has to contain more diversified contents that are relevant to user's query than in the generic query-focused summarization task. We settled on aspects that are pre-defined diverse types of needed information on satisfying user's requests. The pre-defined information aspects depending on query category help to consider covering these information explicitly in our approach. Thus we propose another method to ensure covering several aspects of information needed by users query.

In this paper, we described our systems and the evaluation results at the NTCIR-9 1CLICK task. These systems consist of two different summarization methods, both of which use Integer Linear Programming. The first system adopts QSBP (QuerySnowball with word pair) for query oriented summarization, and extends the method to abstractive summarization. Second system employs Max-min problem to solve sentence extraction problem for extractive summarization. When this system selects sentences, the generated summary covers various informations widely according to the max-min problem. Our first and second systems achieved 0.1585 and 0.1484 S-measure(I) score respectively in the desktop task. Furthermore, our first and second systems achieved 0.0866 and 0.0829 S-measure(I) score respectively in the mobile task.

The rest of this paper is organized as follows. Section 2 describes a brief introduction to Integer Linear Programming we employed, and Section 3 describes our systems. Section 4 describes our experimental settings, and Section 5 describes official results and discussion. Section 6 concludes this paper.

## 2. INTEGER LINEAR PROBLEM

Integer Linear Programming(ILP) has been studied for summarization recently. The problem is a kind of linear programming, that is constrained its variables are trapped in integer. Summarization is often expressed as a maximization of containing information about source document under length constraint. ILP can solve the maximization problem exactly. In contexts of summarization, Carbonell and Goldstein [2] proposed the Maximal Marginal Relevance (MMR)

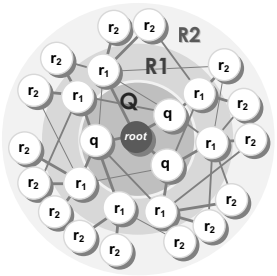


**Figure 1: Co-occurrence Graph (Query Snowball)**

criteria for non-redundant sentence selection, which consist of the document similarity and the redundancy penalty. Then, McDonald [7] presented an approximate dynamic programming approach to maximize the MMR criteria and a formulation of the summarization task as an Integer Linear Problem to provide exact solution. Yih et al. [11] formulated the document summarization problem as an MCKP that is a class of ILP, and proposed a supervised method. Whereas, our method is unsupervised. Filatova and Hatzivassiloglou [3] also formulated summarization as an MCKP. Moreover, it is also used in several NLP tasks [4, 10] .

Although ILP a has great power and a wide application possibility, there are some difficulties in using ILP in practical situations. To solve ILP problems require large computational resources even if ILP problems are a small problem. Computational time becomes exponentially longer, as a problem becomes larger. It is hard to estimate computational cost in advance. In general, GLPK [1] and CPLEX[2] are commonly used to solve ILP problems.

# 3. PROPOSED METHODS

## 3.1 First approach

### 3.1.1 Query Snowball method (QSBP)

The basic idea behind QSB(QuerySnowball) [8] is to close the gap between the query (i.e. information need representation) and relevant sentences by enriching the information need representation based on co-occurrences. To this end, QSB computes a *query relevance score* for each word in the source documents as described below.

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

Our first clue for computing a word score is the query-independent importance of the word. We represent this *base word score* by $s_b(w) = \log(N/tf(w))$ where $tf(w)$ is the term frequency of $w$. We derive the frequency of $w$ from Web Japanese N-gram [5]. Our second clue is the weight propagated from the center of the co-occurrence graph shown in Figure 1. Below, we describe how to compute the word scores for words in $R1$ and then those for words in $R2$.

As Figure 1 suggests, the query relevance score for $r1 \in R1$ is computed based not only on its base word score but also on the relationship between $r1$ and $q \in Q$. To be more specific, let $freq(w, w')$ denote the within-sentence co-occurrence frequency for words $w$ and $w'$, and let $distance(w, w')$ denote the *minimum dependency distance* between $w$ and $w'$: A dependency distance is the path length between nodes $w$ and $w'$ within a dependency parse tree; the minimum dependency distance is the shortest path length among all dependency parse trees of source-document sentences in which $w$ and $w'$ co-occur. Then, the query relevance score for $r1$ can be computed as:

$$s_r(r1) = \sum_{q \in Q} s_b(r1) \left( \frac{s_b(q)}{sum_Q} \right) \left( \frac{freq(q, r1)}{distance(q, r1) + 1.0} \right) \quad (1)$$

where $sum_Q = \sum_{q \in Q} s_b(q)$. It can be observed that the query relevance score $s_r(r1)$ reflects the base word scores of both $q$ and $r1$, as well as the co-occurrence frequency $freq(q, r1)$. Moreover, $s_r(r1)$ depends on $distance(q, r1)$, the minimum dependency distance between $q$ and $r1$, which reflects the strength of relationship between $q$ and $r1$. This quantity is used in one of its denominators in Eq.1 as small values of $distance(q, r1)$ imply a strong relationship between $q$ and $r1$. The 1.0 in the denominator avoids division by zero.

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

$$s_r(r2) = \sum_{r1 \in R1} s_b(r2) \left( \frac{s_r(r1)}{sum_{R1}} \right) \left( \frac{freq(r1, r2)}{distance(r1, r2) + 1.0} \right) \quad (2)$$

where $sum_{R1} = \sum_{r1 \in R1} s_r(r1)$.

### 3.1.2 Word Pairs

Having determined the query relevance score, the next step is to define the summary score. To this end, we use word pairs rather than individual words as the basic unit. This is because word pairs are more informative for discriminating across different pieces of information than single common words. Thus, the word pair score is simply defined as: $s_p(w_1, w_2) = s_r(w_1) s_r(w_2)$ and the summary score is computed as:

$$f_{QSBP}(S) = \sum_{\{w_1, w_2 | w_1 \neq w_2 \text{ and } w_1, w_2 \in u \text{ and } u \in S\}} s_p(w_1, w_2) \quad (3)$$

where $u$ is a textual unit, which in our case is a sentence. Our problem then is to select $S$ to maximize $f_{QSBP}(S)$.

### 3.1.3 Summarization

We derive a summary maximizing this score function using ILP solver. QSB and QSBP are designed to process natural language sentences, and cannot deal with structured information such as table. Then, as a preprocessing, we consider existing words in an information table in Wikipedia as having one co-occurrence each other with a dependency distance:3. The processing will help to acquire information related to query when it is on the information table.

- pre-summarization

[1] It is open source software, available in `http://www.gnu.org/s/glpk/`

[2] This software is faster than GLPK in general. It is commercial package provided by IBM, detailed in `http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/`

Due to the computational difficulty, prior to make a summary through ILP based method, we generated an intermediate summary of the source documents in advance. The process is similar to the approximate prediction proposed by Berg-Kirkpatrick.at.al [1]. We used a simple method to extract candidate sentences on the next ILP stage. We rerank sentence order by following score function:

$$f_{\mathrm{QSBP}}(S) = \sum_{\{wp_{w_k,w\prime_k}|(w_k,w\prime_k)\in u \text{ and } u\in S\}} \mathrm{s_p}(w_k, w\prime_k) \quad (k \le k'). \quad (4)$$

Sentences containing large amount of information irrespective of its length, because we will compress the summary again. When the summarization process is complete, we output the extracted and compressed sentences while preserving the original order.We extract up to 200 sentences for each source documents.

- ILP representation

We can represent the sentence extraction and compression as an ILP problem. We regard a pair of words as the basic unit as a basic unit to score a summary. Our objective function is the summation of the scores for the basic units.

When generating an abstractive summary, we have to deal large amount of constraints. In some cases, the large amount of constraints cause increase computational cost. In fact, we could not solve the ILP with grammatical constraints in time. We therefore omitted the dependency constraint from the problem as follows. This will hurt readability, but may decrease its computational cost.

By solving the following ILP, we can acquire a compressive summary:

maximizing

$$f_{\mathrm{QSBP}}(S) = \sum_{\{wp_{w_k,w\prime_k}|(w_k,w\prime_k)\in u \text{ and } u\in S\}} \mathrm{s_p}(w_k, w\prime_k) \quad (k \le k') \quad (5)$$

subject to

$$cz_{(i,j,j')} \ge c_{i,j}$$
$$cz_{(i,j,j')} \ge c_{i,j'}$$
$$wp_{w_k,w_{k'}} \le \sum_{\{(i,j,j')|w_{(i,j)}=w_k, w_{(i,j')}=w_{k'}\}} cz_{i,j,j'}$$
$$c_{i,j} \in \{0,1\}$$
$$cz_{(i,j,j')} \in \{0,1\}$$
$$wp_{wp_{w_k,w_{k'}}} \in \{0,1\}$$

where $c$ and $cz$ are indicator variables corresponding individual word instance and word pair instance, respectively. $wp$ indicates a class of $cz$ indicating same pair of words.

## 3.2 Second approach

In NTCIR-9 1CLICK, all queries have only one category. For example, a query "Keiko Kitagawa (Japanese actress)" belongs to the CE (celebrity) category. The kinds of information we want to know about her are assumed to be diverse; her birthday, age, interest, or TV dramas. To generate a summary that covers such diverse information about a given query, we defined aspects as follows. An aspect is a type of important information corresponding to a given query category, and there are multiple aspects corresponding to a category. Moreover, aspects are different depending on the categories. In order to capture these aspects, we train classifiers that predict the degree to which sentence reflects each aspect.

The goal of this task is to generate the summary that contains diverse information about query. Therefore, we ensure a summary to contain multiple aspects widely. We used outputs of aspect classifiers to select sentences as a summary. We solve a sentence selection problem as a max-min problem. A max-min problem can be formalized as ILP that we can solve exactly. Our approach generates a summary that covers aspects widely to solve the max-min problem. In the following sections, we explain more details of our approach.

### 3.2.1 Query categorization

At first, we need to classify query into a category, because we defined aspects with respect to each category. There are four categories: "CE (celebrity)", "LO (local)", "DE (definition)", "QA (question answering)" in this task. To classify queries, we used SVM. Since this query categorization problem is a multilabel classification problem, we applied one vs. rest approach. We used the frequencies of bigrams in retrieved web pages as features of the classifier. We created our training datasets for the query classification by crawling query relevant URLs provided as training datasets we will describe in the next section.

### 3.2.2 Labeling training data

We used NTCIR-9 1CLICK training data provided by NTCIR 1CLICK task organizer, that consists of nugget id, nugget semantics, vital string, URL - "*N001*(nugget id), *official blog* `http://star-studio.jp/kitagawa-keiko`(nugget semantics), `star-studio.jp/kitagawakeiko`(vital string), `http://star-studio.jp/kitagawa-keiko`(URL)". Nugget semantics means the meaning of nugget, and is usually represented as a factual sentence. A vital string is an important information of nugget semantics that should be contained in the summary. An URL indicates a web page that contains nugget semantics. We labeled training data for training aspect classifiers. We defined aspects as in Table 1 and Table 2.

| label | size | label | size |
|---|---|---|---|
| real name | 8 | works, publications | 278 |
| birthday | 9 | career | 205 |
| hometown | 9 | web site | 16 |
| contact | 3 | family | 9 |
| job | 22 | physical attribute | 14 |
| interest, personality | 55 | | |

**Table 1: labels of category "CE"**

"*Career*" in Table 1 means his/her educational background, a team or an organization that he or she belongs to or used to belong to. "*Physical attribute*" in Table 1 means his/her height, weight or blood type. "*Interest, personality*" in Table 1 means his/her mind, hobby, favorite foods or least favorite foods. We showed factual labeled training data in Table 3.

We regard the queries categorized to "DE" or "QA" as

| label | nugget semantics |
|---|---|
| real name | *real name Yui Aragaki, real name is Keiko Kitagawa* |
| birthday | *date of birth June 11, 1988, born August 22, 1986* |
| hometown | *born in Hyogo Pref., Naha, Okinawa Pref. the birthplace* |
| contact | *contact 03-3479-xxxx, e-mail adress aaaaa@k-hata.jp* |
| hometown | *born in Hyogo Pref., Naha, Okinawa Pref. the birthplace* |
| job | *actress, occupation ex-NHK official, catcher position, trac and field athlete* |
| interest, personality | *favorite writer Kiyoshi Shigematsu, allergy ultraviolet, hoby wathcing movie, Giants fan, keeping the cat, personality shy* |
| works, publications | *her film Watashi-dasuwa, her TV Dragon-sakura,* |
| career | *enrolled in Meiji University in 2005, graduated from Meiji University in 2009, selected to Miss Seventeen and debut,* |
| web site | *official blog http://www.lespros.co.jp/sample/yui_aragaki/yui_blog, official site http://www.k-hata.jp, twitter http://twitter.com/paserikiri,* |
| family | *divorced 2009, married 2003, husbund Eita,* |
| physical attribute | *height 160cm, blood type O,* |

**Table 3: examples of factual training data**

| label | size | label | size |
|---|---|---|---|
| address | 14 | TEL, FAX | 106 |
| e-mail adress | 11 | access | 62 |
| holiday | 5 | opening hour | 10 |
| contact | 22 | parking | 20 |

**Table 2: labels of category "LO"**

being out of the scope of our system, and did not generate a summary for those queries.

### 3.2.3 Prediction of aspects

We used maximum-entropy classifiers to capture aspects. In this process, classifiers are used to predict whether sentences reflect the aspect or not. A maximum-entropy classifier outputs not a binary, but a real value indicating the probability that the sentence belongs to the class. We regard this output as a degree of aspect's reflection which sentences have. We emphasize that we use a maximum-entropy classifier as not for classification but for prediction of how sentences reflect the aspect. To train the classifier, nugget semantics labeled as the aspect we want to predict is used as positive instances otherwise as negative instances. We train classifiers for each aspect of each category.

### 3.2.4 Max-min problem

Using the results of aspect scorer, we formalize our objective function to select sentences as a summary. Here, we define $s_i$ as a binary variable that takes 1 if we select sentence $i$ as the part of summary, otherwise 0. Let $p_{ij}$ denote the value of aspect $j$ to sentence $i$. Furthermore, we assumed that the score of aspect $j$ of summary is $\sum_i s_i p_{ij}$. Since our goal is to generate the summary that covers multiple aspects widely,

the score of every aspect needs to be large. We can implement this idea in the following maximization problem.

$$\begin{aligned} \text{maximize} \quad & \min_{j \in aspect} \sum_i s_i p_{ij}, \qquad (6) \\ \text{subject to} \quad & \\ & \sum_i s_i l_i \leq L, \\ & s_i \in \{0, 1\}. \end{aligned}$$

Here, we introduce $z$ that is equal to or smaller than any score of aspect $j$ in summary. For this reason, when $z$ is maximized, the minimum score of the aspects in the summary is also maximized. Thus, our objective is to maximize $z$. We can reformalize our objective (6) to new objective (7) due to $z$ and this formalization is a form of ILP.

$$\begin{aligned} \text{maximize} \quad & z, \qquad (7) \\ \text{subject to} \quad & \\ & \sum_i s_i l_i \leq L, \\ & \sum_i s_i p_{ij} \geq z \quad \forall j \in aspects, \\ & s_i \in \{0, 1\}. \end{aligned}$$

## 4. EXPERIMENTS

We clawled source documents from given URL list and removed HTML tags. If the crawler found '*table*' tags, '*th*' and '*td*' in the same '*tr*' tag are combined as a sentence. Because nugget semantics looks like similar to the sentence combined with '*th*' and '*td*'. Especially '*infobox*' in Wikipedia is very beneficial for us.

Unfortunately, some URLs (at least 10 URLs at 2011/9/20) responded '*404 not found*' in ORACLE's URL list.

## 4.1 Query categorization of TTOKU-2

The sizes of training data to classify queries for both TTOKU-M-ORCL-2 and TTOKU-D-ORCL-2 are showen on Table 4.

```
<table class="infobox”>
  <tr>
  <th>手塚 治虫</th>
  </tr>
  <tr>
  <th>本名</th>
  <td>手塚 治</td>
  </tr>
  <tr>
  <th>生誕</th>
  <td>1928年 11月3日</td>
  ...
```

本名　手塚治虫

生誕　1928年 11月3日

**Figure 2: structure of *table* tag**

| category | query num | size [*kb*] |
|---|---|---|
| CE | 11 | 764 |
| LO | 11 | 305 |
| DE | 11 | 1500 |
| QA | 11 | 282 |

**Table 4: sizes of training data for all four categories**

# 5. RESULTS AND DISCUSSION

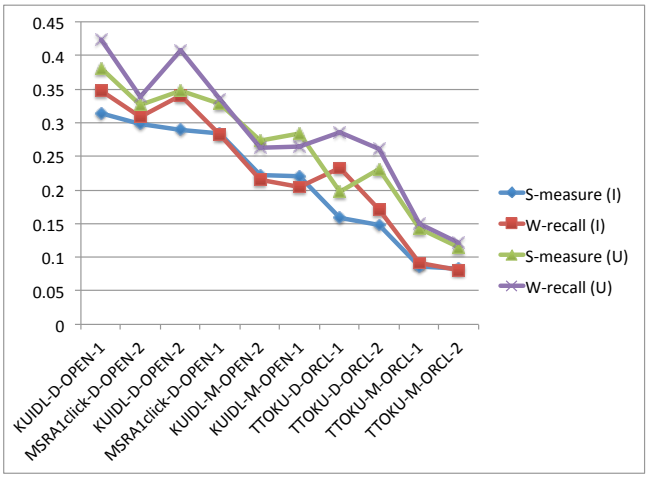


**Figure 3: mean S-measure/W-recall of all systems**

The results are shown in Table 5 and Figure 3. In addition to our systems, the results of other systems those which performed best score at each task both desktop and mobile are also shown in table 5. W-recall is weighted recalll and S-measure is contributed by both weight and position of nuggets matched with gold standard. Even though our runs ar ORACLE runs, our performances are lower than the OPEN runs.

## 5.1 Discussion

For the purpose of comparison within the ORACLE setting, we implemented a state-of-the-art summarization method as a baseline. Below, we discuss the comparative results.

Our two approaches share a common problem, namely, the insufficiency of the web page pre-processing. Probably the most serious problem of our methods is that they do not utilize the web page structure. For example, it is often insufficient to extract just a list of (say) book titles from a web page, because a mere book title will not cover a nugget that represents “Author X wrote the book [booktitle].” That is, we cannot tell whether it is a book title and who wrote it. To solve this, we would also need extract the labels around the list within the web page.

Our approaches also suffered from template description such as ”login” and ”top page”. Poor content extraction affects the quality of our summaries. In terms of extraction units, we extracted descriptions in a unit of a sentence and a morpheme. In general, we have to wrap proper noun or address and zip code into one unit, and should not separate them for sentence extraction or sentence compression.

### 5.1.1 *Baseline*

Hui ling and Brimes[6] designed a monotone submodular function for query-oriented summarization. Their succinct method achieved good performance in DUC from 04 to 07. They proposed positive diversity reward function for a non-redundant summary in order to define a monotone submodular objective function for generating summary. The diversity reward gives smaller gain for a biased summary, because it consists of $C_k$ parts of cluster $c$, and calculates a square rooted score with respect to each sentence. The reward also contains similarity to a query when try to generate query-oriented summary. Their objective function also includes a coverage function based on the similarity $w_{i,j}$ between sentences. In the coverage function min function limits its maximum gain $\alpha \sum i \in V w_{i,j}$ that is a small fraction $\alpha$ of similarity between a sentence $j$ and whole source document. The objective function is sum of the positive reward $\mathcal{R}$ and the coverage function $\mathcal{L}$ over source documents $V$, as follows:

$$\mathcal{F}(S) = \mathcal{L}(S) + \sum_{k=1}^{3} \lambda_k \mathcal{R}_{Q,k}(S), \quad (8)$$

$$\mathcal{L}(S) = \sum_{i \in V} \min \left\{ \sum_{j \in S} w_{i,j}, \alpha \sum_{k \in V} w_{i,k} \right\},$$

$$\mathcal{R}_{Q,k} = \sum_{c \in C_k} \sqrt{\sum_{j \in S \cup c} \left( \frac{\beta}{N} \sum i \in V w_{i,j} + (1-\beta) r_{j,Q} \right)},$$

where $\alpha$, $\beta$ and $\lambda_k$ are parameter, $r_{j,Q}$ represents the similarity between sentence $j$ and query $Q$. They used three cluster $C_k$ with different granularity (0.2N,0.15N,0.05N), that is calculated in advance.

Originally, they developed parameters using grid search on training data. In this case, we need to evaluate summaries manually, the grid search come with difficulty. Therefore, we set parameters ($\alpha = \frac{5}{N}$,$\beta = 0.5$,$\lambda_{\{1,2,3\}} = 6$) by reference to their result on DUC-03 [6]. We set stopwords to “教える (teach)”,“知る (know)”,“何 (what)” and their variation, that is characteristic for a question query. For the query expansion, we used Japanese wordnet to obtain synonym and hypernym of query terms.

Table 6 shows a result of evaluation of the baseline. In this task every run was evaluated by two assessor, but only one native Japanese evaluated this run. The baseline is locates

| | S-measure(I) | W-recall(I) | S-measure(U) | W-recall(U) |
|---|---|---|---|---|
| KUIDL-D-OPEN-1 | 0.3132 | 0.3468 | 0.3814 | 0.4236 |
| TTOKU-D-ORCL-1 | 0.1585 | 0.2321 | 0.1969 | 0.2851 |
| TTOKU-D-ORCL-2 | 0.1484 | 0.1704 | 0.2316 | 0.2610 |
| KUIDL-M-OPEN-2 | 0.2214 | 0.2147 | 0.2730 | 0.2624 |
| TTOKU-M-ORCL-1 | 0.0866 | 0.0921 | 0.1418 | 0.1493 |
| TTOKU-M-ORCL-2 | 0.0829 | 0.0779 | 0.1312 | 0.1211 |

**Table 5: mean S-measure/W-recall**

between our intersection score and union score, considering the difference experiment settings, the baseline are comparable with or slightly better than our method. Looking at the results by query types shown in table 7, our approaches perform better than this results in DE, CE and LO queries. Meanwhile, in QA queries, the baseline performs much better. This is because our approaches do not use query terms directly, and many answer nuggets of QA types tend to co-occur with query terms, indirect usage might be inefficient to collect directly co-occurring answer nuggets. Compared to the runs from other teams, our baseline also performs poorly. While the other runs are OPEN runs and our runs (including the baseline) are ORACLE runs, ours use summarization techniques only. Thus, in addition to web page pre-processing, we will probably need to adopt question answering and information retrieval techniques to boost our performances.

### 5.1.2 First approach

Our first approach, TTOKU-D(M)-ORCL-1, performed particularly badly in terms of readability. The readability was evaluated within the range between -2 to 2, and our result is -0.75(-0.85). The biggest factor is that we omitted dependency constraints in sentence compression. A second factor may be that the sentences we extracted were too long, which resulted in over-compression and therefore poor readability.

Moreover, QSBP probably had trouble handling web pages and Wikipedia. As was discussed earlier, the web page structure often separates words into different segments even though the words have an important relationship to each other. In addition, the symbols that often appear in Wikipedia (e.g. "∧" that indicates footnote ) tend to co-occur with many words and will be treated as important by QSBP.

### 5.1.3 negative effect of sentence compression

Sentence compression can remove unnecessary description and shorten the length of summary. On the other hand, it may also have negative effects for covering nuggets by removing vital information from the sentences. There are cases where a sentence covered a nugget but was unreadable due to compression, as well as those where the entire nugget was deleted from a relevant sentence due to compression. Table 8 shows the effects of sentence compression on randomly chosen 10 queries from TTOKU-ORCL-D-1.

Syntactic constraints are required to improve the readability of compressed sentences. To balance the readability and content informativeness is a challenging task. To improve the effectiveness of our approach, we will have to determine more accurately when we can remove non-vital information without hurting readability. Moreover, this needs to be accomplished efficiently.

| | |
|---|---|
| lost by deletion | 39(14.8%) |
| lost by poor readability | 8(3.0%) |
| total nuggets | 264 |
| compression rate | 57.3% |

**Table 8: Effects of sentence compression in TTOKU-D-ORCL-1**

### 5.1.4 second approach

The performance of the query categorization is not good. The rate of true positive out of total queries is 17/60. CE, DE and QA are sometimes mislabeled as LO. As query categorization was not the focus of our study, we did not spend time on feature selection and parameter tuning.

Next, we analyze the performance of the maximum-entropy classifiers whether they can predict the score of the aspect. At first, we calculate the average of the scores of the aspect for each training nugget. Then, we calculate the average of the scores of the aspect for each training nugget where the training nugget is positive instance. The proportion of these two score is larger than 1.0. This means the maximum entropy classifier predicts the score of the aspect appropriately. Usually, the evaluation metrics of the classification are accuracy, precision, recall and F-measure. But in our second approach, we need to differentiate between the scores of positive instances and those of negative instances.

Therefore, the nuggets are categorized successfully. However, there is no guarantee that a nugget exists in the source documents.

The range of the score of the aspects are different depending on the size of the training data. Thus, we should normalize the scores of the aspects.

Our current approaches do not include any special treatments of DE and QA queries. It may be worthwhile to extend our model by incorporating a factor for query-sentence similarity into our objective function.

## 6. CONCLUSION

Our first and second systems achieved 0.1585 and 0.1484 S-measure(I) scores respectively in the desktop task. Furthermore, our first and second systems achieved 0.0866 and 0.0829 S-measure(I) score respectively in the mobile task.

Although our runs are ORACLE runs (i.e. used the supporting URLs as input to our summarizers), there is a lot of room for improvement in our results. We also implemented a state-of-the-art baseline. Then we conducted failure analyses. We are planning to bring rich constraints for com-

| | S-measure(I) | S-measure(U) | W-recall(I) | W-recall(U) |
|---|---|---|---|---|
| TTOKU-D-ORCL-1 | 0.1585 | **0.1969** | **0.2321** | **0.2851** |
| TTOKU-D-ORCL-2 | 0.1484 | **0.2316** | 0.1704 | **0.2610** |
| Hui Ling.et.al(2011) | 0.1706 | | 0.2215 | |

**Table 6: mean S-measure/W-recall**

| | | S-measure(I) | S-measure(U) | W-recall(I) | W-recall(U) |
|---|---|---|---|---|---|
| CE (celebrity) | TTOKU-D-ORCL-1 | 0.0215 | 0.0401 | 0.0156 | 0.0297 |
| | TTOKU-D-ORCL-2 | **0.1276** | **0.2100** | **0.0881** | **0.1414** |
| | Hui Ling.et.al(2011) | 0.0578 | | 0.0592 | |
| LO (local) | TTOKU-D-ORCL-1 | **0.1207** | **0.1623** | **0.1957** | **0.2409** |
| | TTOKU-D-ORCL-2 | **0.0972** | **0.1165** | 0.0905 | **0.1033** |
| | Hui Ling.et.al(2011) | 0.0945 | | 0.0961 | |
| DE (definition) | TTOKU-D-ORCL-1 | **0.2005** | **0.2728** | **0.3266** | **0.4489** |
| | TTOKU-D-ORCL-2 | **0.2208** | **0.2927** | **0.2905** | **0.4050** |
| | Hui Ling.et.al(2011) | 0.1534 | | 0.2392 | |
| QA (question answering) | TTOKU-D-ORCL-1 | 0.2915 | 0.3123 | 0.3906 | 0.4211 |
| | TTOKU-D-ORCL-2 | 0.0977 | 0.2487 | 0.1278 | 0.2883 |
| | Hui Ling.et.al(2011) | 0.3764 | | 0.4914 | |

**Table 7: mean S-measure/W-recall on by query type**

pression, and generate readable summary. We also plan to extend second methods to treat query directly.

# 7. REFERENCES


[1] T. Berg-Kirkpatrick, D. Gillick, and D. Klein. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 481–490, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[2] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM.

[3] E. Filatova and V. Hatzivassiloglou. A formal model for information selection in multi-sentence text extraction. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[4] R. Iida and M. Poesio. A cross-lingual ilp solution to zero anaphora resolution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 804–813, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

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

[6] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 510–520, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

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

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

[9] S. Tetsuya, M. P. Kato, and Y. I. Song. Overview of ntcir-9 1click [draft]. 2011.

[10] K. Thadani and K. McKeown. Optimal and syntactically-informed decoding for monolingual phrase-based alignment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 254–259, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[11] W. T. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. Multi-document summarization by maximizing informative content-words. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1776–1782, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.