TUTA1 at the NTCIR-11 IMine Task

Hai-Tao Yu
The University of Tokushima
yu-haitao@iss.tokushima-u.ac.jp

Fuji Ren
The University of Tokushima
ren@is.tokushima-u.ac.jp

ABSTRACT

In this paper, we detail our participation in two subtasks: *subtopic mining* and *document ranking* of the NTCIR-11 IMine task. In the subtopic mining subtask, to discover the latent hierarchy among query-like strings, our key idea is to structurally parse query-like strings by characterizing pairwise dependency in the bag-of-units perspective. Then the clustering algorithm (i.e., affinity propagation) and the Sainte-Laguë algorithm are used to obtain the target list that represents a two-level hierarchy of subtopics. In the document ranking subtask, we deploy the newly proposed *0-1 MSKP* model for diversified document ranking against unclear topics. A subset of documents are optimally chosen like filling up multiple subtopic knapsacks.

Team Name

TUTA1

Subtasks

Subtopic Mining (Chinese, English)
Document Ranking (Chinese, English)

Keywords

Subtopic; Diversification; 0-1 MSKP

1. INTRODUCTION

The TUTA1 group at The University of Tokushima participated in two subtasks of the NTCIR-11 IMine task, i.e., subtopic mining and document ranking. For detailed information of each subtask, please refer to the overview paper [10]. In the subtopic mining subtask, different from precious "subtopic mining" subtasks of NTCIR-9 and NTCIR-10, this time a two-level hierarchy of possible subtopics is expected for *unclear* topics. But the most knotty problem remains the same, namely, query-like strings generally are short and show no grammatical or syntactical structure. It is hard to capture the encoded information need or intent. On the basis of *bag-of-units* perspective (a unit can be a word or phrase), we structurally parse query-like strings by characterizing pairwise dependency. This method enables us to identify equivalent subtopic strings and quantify the similarity between a pair of subtopic strings in a new way. Once the similarities among candidate subtopic strings are quantified, the affinity propagation algorithm [5] can be deployed to generate the two-level hierarchy of possible subtopics against unclear topics. And the Sainte-Laguë method[1] can be used to obtain the target list of representative subtopic strings.

The document ranking subtask uses the same topics as the subtopic mining subtask. Based on the subtopic mining results, participants are asked to selectively perform diversified document ranking. To this end, we experiment with our newly proposed 0-1 MSKP model [14] for search result diversification w.r.t. unclear topics.

In the remainder of this paper, we outline the notations, the methods for intermediate natural language parsing in §2. §3 and §4 detail the approaches proposed for subtopic mining and document ranking subtasks respectively. We conclude our work in §5.

2. PRELIMINARIES

In this section, we first outline the notations used throughout this paper, then detail the preliminaries for both subtopic mining and document ranking subtasks.

Let t denote a topic, st denote a possible *subtopic*. Here the concept of subtopic refers to a possible information need or an intent underlying a topic. A *subtopic string*, denoted as $stStr$, is viewed as an expression of a subtopic. Consider the well-worn topic *Harry Potter* for example, *Harry Potter fiction* and *Harry Potter reading* are regarded as two subtopic strings about the subtopic *book* (i.e., book-centric information needs, here book is used for short). Query-like strings, such as query suggestions collected from web search engines and text segments extracted from relevant documents or

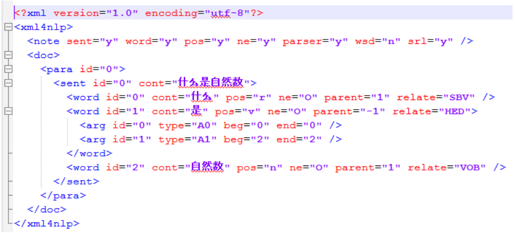
[1] http://en.wikipedia.org/wiki/Sainte-Laguë _method

other resources, are used as *raw subtopic strings*. The reasons why we call them raw subtopic strings are that: (1) Some of them may be noisy ones and have little or no use. (2) Some are equivalent and represent a similar subtopic. Thus preprocess is necessary in our perspective. Different from English that uses a space to separate words, the Asian languages like Chinese are not separated via any punctuation or spaces. A word in the Asian languages like Chinese generally refers to a semantic unit, which is also different from the "word" in English. Therefore, the concept of term is used to refer to a semantic unit at a word granularity.

2.1 Natural Language Parsing

2.1.1 *Chinese Natural Language Parsing*

In the subtopic mining subtask, LTP-Cloud[2] is used to perform linguistic annotation. For a piece of text, the pipeline functionality enables sentence splitting, tokenization, part-of-speech (POS) tagging, named entity recognition (NER), dependency parsing, word sense disambiguation and semantic role labeling. Fig.1 shows the annotation result of topic 0037: 什么是自然数(what is natural number). For detailed description of each tag, please refer to the documentation of LTP-Cloud.



```
<?xml version="1.0" encoding="utf-8"?>
<xml4nlp>
  <note sent="y" word="y" pos="y" ne="y" parser="y" wsd="n" srl="y" />
  <doc>
    <para id="0">
      <sent id="0" cont="什么是自然数">
        <word id="0" cont="什么" pos="r" ne="O" parent="1" relate="SBV" />
        <word id="1" cont="是" pos="v" ne="O" parent="-1" relate="HED">
          <arg id="0" type="A0" beg="0" end="0" />
          <arg id="1" type="A1" beg="2" end="2" />
        </word>
        <word id="2" cont="自然数" pos="n" ne="O" parent="1" relate="VOB" />
      </sent>
    </para>
  </doc>
</xml4nlp>
```

Figure 1: Linguistic annotation for topic 0037.

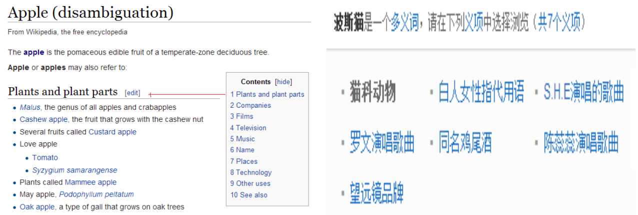
Based on the dependency parsing result of a Chinese query-like string, if the grammatical relations subject-verb (tagged as SBV) and verb-object (tagged as VOB) exist at the same time (e.g., Fig.1), it is regarded as a complete sentence. Based on the POS tagging result of a Chinese query-like string, two consecutive noun terms are regarded as a *noun phrase*. To reduce the impact of *out-of-vocabulary* (OOV) terms, e.g., the Chinese topic 0001: 先知(a polysemic term), we added the *rule-1* that a piece of text consisting of no more than 3 Chinese characters is directly regarded as a term. In the document ranking subtask, ICTCLAS2014[3] is used to perform word segmentation.

2.1.2 *English Natural Language Parsing*

For English natural language parsing, Stanford CoreNLP[4] is used. For a piece of text, the pipeline functionality includes sentence splitting, tokenization, POS tagging, lemmatization, NER, syntactic parsing and coreference resolution. Based on the syntactic parsing result of a query-like string, it is regarded as a complete sentence when either *rule-2* or *rule-3* is met: (1) rule-2, nominal subject (tagged as nsubj) exists and the identified copula is not a possessive form (i.e., "'s"). (2) rule-3, direct object (tagged as dobj) and adverbial modifier (tagged as npadvmod) exist at the same time.

2.2 Online Knowledge Extraction

Wikipedia[5] is known as the largest online English encyclopedia, like Baike[6] for Chinese. They contain millions of entries, most of which are named entities, keyword phrases throughout a large number of domains. Corresponding to each entry, the well-written encyclopedia article provides a notable encyclopedic topic, summarizes this topic comprehensively, contains references to reliable sources, and links to other related topics. What is more, for the case that two or more different topics could have the same entry (say, a polysemic entry), the *disambiguation page* is provided by Wikipedia (e.g., Fig.2(a) for topic 0051: apple[7]), and a similar page by Baike (e.g., Fig.2(b) for topic 0002: 波斯猫[8]). Fig.2(a) and Fig.2(b) show that the disambiguation pages by human intelligence summarize polysemic entries comprehensively.



Apple (disambiguation)

From Wikipedia, the free encyclopedia

The **apple** is the pomaceous edible fruit of a temperate-zone deciduous tree.

Apple or **apples** may also refer to:

Plants and plant parts [edit]

- *Malus*, the genus of all apples and crabapples
- Cashew apple, the fruit that grows with the cashew nut
- Several fruits called Custard apple
- Love apple
  - Tomato
  - *Syzygium samarangense*
- Plants called Mammee apple
- May apple, *Podophyllum peltatum*
- Oak apple, a type of gall that grows on oak trees

Contents [hide]
1 Plants and plant parts
2 Companies
3 Films
4 Television
5 Music
6 Name
7 Places
8 Technology
9 Other uses
10 See also

(a) Segment from Wikipedia

波斯猫是一个多义词，请在下列义项中选择浏览（共7个义项）

- 猫科动物
- 白人女性指代用语
- S.H.E演唱的歌曲
- 罗文演唱歌曲
- 同名鸡尾酒
- 陈慈慈演唱歌曲
- 望远镜品牌

(b) Segment from Baike

Figure 2: Segments of the disambiguation page.

In this study, for each topic that has a disambiguation page, the subtopic strings representing different subtopics are collected. For example, "Plants and plant parts" and "Companies" in Fig.2(a), "猫科动物" and "望远镜品牌" in Fig.2(b). Meanwhile, subtopic strings representing a similar subtopic are further merged if one or more common terms

[2] http://www.ltp-cloud.com/intro/en/
[3] http://ictclas.nlpir.org/
[4] http://nlp.stanford.edu/software/corenlp.shtml
[5] http://en.wikipedia.org/wiki
[6] http://baike.baidu.com/
[7] http://en.wikipedia.org/wiki/Apple_(disambiguation)
[8] http://baike.baidu.com/subview/2861/5036145.htm

exist for a pair of subtopic strings. For example, “S.H.E演唱的歌曲” and “罗文演唱歌曲” in Fig.2(b) will be merged to indicate the same subtopic because of the common terms “演唱” and “歌曲”.

3. SUBTOPIC MINING

3.1 Equivalent Subtopic Strings

On the basis of bag-of-units perspective, we structurally parse query-like strings by characterizing pairwise dependency. The concepts of *kernel-object* and *modifier* proposed by Yu and Ren [13] are used to: (1) emphasize the role of each composing unit; (2) characterize the pairwise dependency of composing units. Specifically, a query-like string is viewed as a bag of terms. Kernel-object refers to the dominant term that abstracts the core object or topic of the underlying subtopics. Modifier refers to the other co-appearing terms with kernel-object, which explicitly specify users' interested attributes or concrete aspects. Take the query “哈利波特游戏攻略(Harry Potter game guide)” for example, “哈利波特(Harry Potter)” has a larger probability to be submitted as a kernel-object rather than a modifier. On the contrary, “游戏(game)” and “攻略(guide)” have larger probabilities to be co-appearing modifiers. For queries like “ 萧潇出演过什么电视剧(the TV plays that XiaoXiao has starred in)” and “培养孩子读书习惯(cultivate children's reading habits)”, Yu and Ren [13] treated them as *role-implicit* queries which can not be parsed with kernel-object and modifier. And the concept of *role-explicit* query is used to refer to queries that can be parsed with kernel-object and modifier. For a role-explicit query, there must be one and only one kernel-object, and the number of modifiers is not limited. Moreover, they assume that modifiers are dependent on kernel-object, and modifiers are mutually independent. The underlying intuition is that: When forming an information need, users firstly conceive the kernel-object, the co-appearing modifiers are conceived subsequently and (kernel-object)-dependent.

In our work, we perform two types of structural annotation. (1) *term-level structural annotation*, namely, the kernel-object and modifier are annotated based on the bag-of-terms model. (2) *phrase-level structural annotation*, namely, the kernel-object and modifier are annotated based on the bag-of-phrases model. For an unclear topic, when performing term-level/phrase-level structural annotation, each unique noun term/phrase is equally regarded as the kernel-object, the other terms are viewed as co-appearing modifiers (stop terms are not considered). For an unclear topic including different noun terms/phrases, there will be multiple term-level/phrase-level structural annotations. Furthermore, the structural annotations of an unclear topic are used as references when performing structural annotation for relevant subtopic strings. Corresponding to each structural annotation of the topic, the same term/phrase (w.r.t. the kernel-object of the topic) in a subtopic string is selected as the kernel-object, the other co-appearing terms/phrases are regarded as modifiers. If a subtopic string does not include the same term/phrase as the kernel-object of the topic, a *null* annotation will be recorded. By this way, the subtopic strings have the same number of term-level/phrase-level structural annotations as the unclear topic, on which the similarity between a pair of subtopic strings can builds (§3.2).

Among the raw subtopic strings of a topic, we find that some subtopic strings express a similar subtopic and can be identified literally. E.g., “东风日产阳光最低报价” and “日产阳光最低报价是多少” for the topic 0014: 阳光. We identify subtopic strings of this kind based on their structural annotations of the same type. Namely, if they share the same kernel-object and two modifiers at least, two subtopic strings are regarded as *equivalent subtopic strings*. This merging task is performed as a preprocess. The input subtopic strings of the clustering algorithm (§3.2) are essentially representative ones of a group of equivalent subtopic strings.

3.2 Strategy for Subtopic Mining

For the subtopic mining subtask, a two-level hierarchy of possible subtopics is expected. To this goal, our intuitive strategy is that: Given a set of sufficient subtopic strings w.r.t. a topic, we generate a set of clusters. A cluster consisting of a star subtopic string and several satellite subtopic strings is used to denote a two-level subtopic. The star subtopic string represents the first level subtopic, the satellite subtopic strings represent the second level subtopic. Once a well organized set of clusters is obtained, we use the Sainte-Laguë method to generate the target ranked list for subtopic mining. The Sainte-Lagu ë method is a highest quotient technique for allocating seats in the parliament to members of competing political parties, whilst respecting the requirement that the number of seats a party possesses is proportional to the number of votes it has received. Dang and Croft [4] studied how to use a similar idea to perform diversified document ranking. Analogously, we view each position in the target list as a seat, each cluster as a party, the popularity of a cluster as its votes. Algorithm 1 shows how to generate the target list R. p_k represents the popularity of cluster c_k and is used as the votes that c_k receives, h_k represents the seats that have been assigned to c_k so far. st^* and st represent a star subtopic string and a satellite subtopic string respectively.

Algorithm 1 The Sainte-Laguë method for list generation.

```
$h_k = 0, \forall k$
$j = 1$
repeat
    for all $c_k$ do
        $quotient[k] = \frac{p_k}{h_k + 1}$
    end for
    $i \leftarrow \mathrm{argmax}_k \, quotient[k]$
    $st \leftarrow$ select the best representative satellite subtopic string $st \in c_i$
    generate the $j$-th record of $R$ with $st^* \in c_i$ and $st$
    $c_i \leftarrow c_i \setminus \{st\}$
    $h_i = h_i + 1$
    $j++$
until $j > |L|$
```

By now, the issue to be addressed is how to generate the well organized set of clusters. According to the type of a given topic, we generate the set of clusters as follows:

1. *Clear topic.* If a topic is a completed sentence (§2.1), it is regarded as a clear topic, a cluster consisting of the topic itself is generated.

2. *Polysemic topic.* If a topic is a polysemic entry, the subtopic strings extracted from the corresponding disambiguation page are used as star subtopic strings, i.e., representing different subtopics (§2.2). Corresponding to each star subtopic string, we form a cluster. The query suggestions and/or related queries are added as satellite subtopic strings into a cluster whose star subtopic string shares the most common terms. For a polysemic topic, we assume uniform popularity for all clusters.

3. *Ambiguous and/or underspecified topic.* Except the topics of the above two types, the remaining topics are regarded as ambiguous and/or underspecified topics. The parameter-free clustering algorithm affinity propagation (AP) is used to generate the set of clusters. As stated in §3.1, the candidate subtopic strings (query suggestions and related queries) are structurally annotated by taking the structural annotations of a topic as references. The similarity between a pair of subtopic strings is derived as follows:

Based on the co-appearing modifiers, function f computes the similarity between pointwise structural annotations, namely the respective structural annotations of two subtopic strings corresponding a specific structural annotation of the topic.

$$f(V_1, V_2) = \frac{\sum_i \sum_j sim(v_1^i, v_2^j)}{|V_1| + |V_2|} \tag{1}$$

where V_1 and V_2 represent the set of co-appearing modifiers of each subtopic string respectively, function *sim* computes the similarity between two terms/phrases (detailed in §3.3.1). Then the averaged similarity of term-level/phrase-level structural annotations is computed as:

$$h(L_1, L_2) = \frac{\sum_{j=1}^{k} f(L_1^j, L_2^j)}{k} \tag{2}$$

where L_1 and L_2 represent two lists of term-level/phrase-level structural annotations. Finally, the similarity between a pair of subtopic strings is a linear combination of the averaged similarity derived from term-level structural annotations and the averaged similarity derived from phrase-level structural annotations:

$$\eta \times h(TL_1, TL_2) + (1 - \eta) \times h(PL_1, PL_2) \tag{3}$$

where TL_1 and TL_2 represent the lists of term-level structural annotations respectively, PL_1 and PL_2 represent the lists of phrase-level structural annotations respectively. η is a trade-off parameter.

Given the clusters generated with the AP algorithm, the star subtopic string is the *exemplar* subtopic string, the popularity of a cluster is computed as the normalized ratio of its composing members. The representative satellite subtopic string is selected based on the number of instances.

3.3 Experiments

3.3.1 Experimental Setup

In the subtopic mining subtask, for English topics, the provided query suggestions are used as raw subtopic strings. For Chinese topics, the provided query suggestions and related queries are used as raw subtopic strings. The parameter η (§3.2) is set as 0.4, which means that the phrase-level structural annotation is biased. As for the affinity propagation algorithm, the damping factor is set as 0.5, the maximum iteration threshold is $20,000$, the message-passing procedure will be terminated after the local decisions stay constant for 20 times of iterations. For computing the similarity between a pair of terms/phrases, the algorithms [9] and [6] are used for Chinese and English respectively. Finally, we submitted one run for Chinese and English topics respectively, i.e., *TUTA1-S-C-1A* and *TUTA1-S-E-1A*.

3.3.2 Experimental Results

Among the Chinese/English topic set, the official numbers of clear topics, broad topics and ambiguous topics are the same, i.e., 17, 17 and 16. The broad and ambiguous topics are viewed as unclear topics. In our study, 3 Chinese topics/(2 English topics) are determined as clear, 47 Chinese topics/(48 English topics) are determined as unclear.

Using the metrics of *precision* and *recall*, the topic classification results is shown in Table 1, where Ch represents the Chinese topics, En represents the English topics.

	Clear topic		Unclear topic	
	Precision	Recall	Precision	Recall
Ch	$\frac{2}{3}$|66.67%	$\frac{2}{17}$|11.77%	$\frac{32}{47}$|68.09%	$\frac{32}{33}$|96.97%
En	$\frac{2}{2}$|100%	$\frac{2}{17}$|11.77%	$\frac{33}{48}$|68.75%	$\frac{33}{33}$|100%

Table 1: Topic classification.

Based on the study [2] claiming that "almost any query could be considered underspecified to some extent", we merely treat a topic being a complete sentence as a clear topic. Thus the recall of clear topic is extremely low. Among the 3 clear Chinese topics by us, the failed one is 0031:相亲节目有哪些(what dating shows are there). It is determined as a complete sentence in our study, while it is officially classified as a broad topic. However, clear topics are not evaluated because they are not expected to contain subtopics. Thus the high recall of unclear topics enables us to get most of the unclear topics.

To evaluate the list that indicates a two-level hierarchy of subtopics, the $H-measure$ (detailed in paper [10]) consisting of $Hscore$, $Fscore$ and $Sscore$ is used. In particular, Fscore/Sscore is defined to measure the quality of the first-level/second-level subtopics in terms of $D\#-nDCG$ [12] respectively. Hscore is defined to measure the quality of the hierarchical structure by whether the second-level subtopic is correctly assigned to the appropriate first-level subtopic. Table 2 shows the subtopic mining performance over unclear topics, where the top-one runs (from the overview paper [10]) in terms of H-measure are indicated in bold.

Run	Hscore	Fscore	Sscore	H-measure
C-top-one run	0.5413	0.5736	0.6339	**0.3360**
TUTA1-S-C-1A	0.2419	0.3242	0.4391	0.1126
E-top-one run	0.9190	0.5670	0.5964	**0.5627**
TUTA1-S-E-1A	0.1933	0.2833	0.3647	0.0688

Table 2: Subtopic mining performance over unclear topics.

As shown in Table 2, our submitted runs perform substantially worse compared with the top-one runs. We identify first-level and second-level subtopics using a specific clustering algorithm. Besides the quality of a clustering algorithm itself, the input affinity matrix representing similarities among subtopic strings is a fundamental issue. The low performance in terms of Hscore demonstrates that: our method that quantifies the similarities among subtopic strings based on the proposed structural parsing is not effective as expected to discover the latent hierarchy among subtopic strings. If the second-level subtopic is not correctly assigned to the appropriate first-level subtopic, small values of Fscore and Sscore will be obtained. Moreover, Hscore is further used as a multiplier to compute the H-measure, thus a smaller Hscore will result in a smaller H-measure value.

4. DOCUMENT RANKING

4.1 Model for Diversified Document Ranking

For document ranking, we deploy the parameter-free 0-1 MSKP model for diversified document ranking w.r.t. unclear topics. Under the 0-1 MSKP model, each possible subtopic is viewed as a knapsack, the number of documents that can be assigned to a subtopic as its capacity, the weight of a document as a unit 1, the relevance score between a document and a subtopic as the profit when assigning documents to subtopic knapsacks. For a topic t to be diversified, let $D = \{d_1, d_2, ..., d_m\}$ be the top-m documents of an initial retrieval run, S be the desired subset of D for forming the result list (commonly, $|S| \ll |D|$), $ST = \{st_1, ..., st_n\}$ be the set of possible subtopics, the popularity of each subtopic be $p_1, ..., p_n$, the task of diversified document ranking, i.e., selecting the optimal subset S of D that are both diverse and relevant w.r.t. ST, is formalized as the following integer linear program:

$$\max \ \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}r_{ij} + \sum_{i=1}^{n} f_i(u_i) + \sum_{j=1}^{m} w_j(x_{:j}, y_j) \tag{4}$$

$$s.t. \ \ y_j = \sum_{i=1}^{n} x_{ij}, \ j \in \{1, ..., m\} \tag{5}$$

$$x_{ij}, y_j \in \{0, 1\}, \ i \in \{1, ..., n\}, j \in \{1, ..., m\} \tag{6}$$

$$u_i = \sum_{j=1}^{m} x_{ij}, i \in \{1, ..., n\} \tag{7}$$

$$\sum_{i} c_i = \sum_{i} u_i, i \in \{1, ..., n\} \tag{8}$$

$$f_i(u_i) = \begin{cases} 0 & , if \ u_i \le c_i \\ -e^{u_i - c_i} & , if \ u_i > c_i \end{cases} \quad , i \in \{1, ..., n\} \tag{9}$$

$$\begin{aligned} w_j(x_{:j}, y_j) = & y_j[\sum_{i \neq k}(1 - s_{ki})(p_k r_{kj} - p_i r_{ij})(x_{kj} - x_{ij}) \\ & + p_k r_{kj} x_{kj}] \ where \ x_{kj} = \max_i x_{ij} \end{aligned} \tag{10}$$

where y_j indicates whether document d_j is selected, x_{ij} indicates whether document d_j is assigned to the subtopic knapsack st_i, r_{ij} denotes the relevance score between subtopic st_i and document d_j, s_{ki} denotes the similarity between the k-th subtopic and i-th subtopic ($s_{ki} \in [0, 1]$), integer variable $u_i(u_i \geq 0)$ denotes the number of documents assigned to subtopic knapsack st_i, and c_i denotes the capacity of subtopic knapsack st_i. For convenience, $x_{:j} = \{x_{1j}, ..., x_{nj}\}$ and $x_{i:} = \{x_{i1}, ..., x_{im}\}$. Correspondingly, $|D| = m$, $|ST| = n$ and $|S| = \sum_i c_i$, i.e., the total weight in the subtopic knapsacks is exactly the size of S. In order to achieve the optimal diversification result, Equation 8, Equations 9 and 10 are ad-hoc restrictions. Specifically,

(1) Instead of hard-capacitated knapsack packing like the traditional 0-1 multiple knapsack problem, Equation 9 means that: a subtopic knapsack will be penalized by function f when the number of assigned documents exceeds its capacity. Combined with Equation 8, they enforce the constraint that the total number of packed documents must be equal to $|S|$.

(2) Equation 10 means that: we prefer the assignment that packing a document into a more relevant subtopic knapsack with a higher popularity. Specifically, under Equation 5 and Equation 6, a document can only be packed into one subtopic knapsack if it is selected. In Equation 10, for a specific document d_j, the product of p_i and r_{ij} represents the relative *profit* if it is assigned to the i-th subtopic knapsack. Once a document d_j is selected (i.e., $y_j = 1$), $x_{kj} = \max_i x_{ij}$ and $x_{kj} - x_{ij}$ will be 1, and x_{kj} also indicates that the host knapsack is the k-th subtopic knapsack. If $p_k r_{kj}$ is not the maximum one in $\{p_1 r_{1j}, ..., p_n r_{nj}\}$, there will be negative values (e.g., $p_k r_{kj} - p_t r_{tj}$, where $p_t r_{tj} > p_k r_{kj}$), which decrease the objective (Equation 4). As s_{ki} represents the similarity between two subtopics, $1 - s_{ki}$ indicates the divergence between two subtopics. By multiplying $1 - s_{ki}$, packing a document into a similar subtopic knapsack is a better choice if not the most profitable one.

Finally, the objective (Equation 4) is to maximize the sum of all relevance scores between the selected documents and their host subtopic knapsacks plus the penalties for knapsacks of which the capacity limit are exceeded, whilst respecting the ad-hoc restrictions. To find the optimal configuration of matrix $\mathbf{x} = [x_{ij}]_{n \times m}$ and vector $\mathbf{y} = [y_j]_{1 \times m}$ that maximizes the objective, the belief propagation algorithm [3, 8] is used. Please refer to paper [14] for detailed information.

4.2 Experiments

4.2.1 Experimental Setup

As for the 0-1 MSKP model, the damping factor is set as 0.5, the maximum iteration threshold is $5,000$, the message-passing procedure will be terminated after the local decisions stay constant for 10 times of iterations. We assume uniform popularity for possible subtopics of an unclear topic, the reason is that the popularity of a subtopic derived merely from query suggestions and/or related queries is not reliable to some extent. As for the models for computing similarity between documents and subtopics, the relevance between subtopics, we follow the same settings as the study [14]. For the initial retrieval run, the provided Chinese baseline is used (no baseline for topic: 0033). For English topics, we perform the initial retrieval over *ClueWeb12-B13* via the search interface provided by Lemur project[9]. Following the default settings, each topic is directly used as a search query and the top-100 documents are used (we failed to get the documents for topics 0076, 0092 and 0100).

4.2.2 Runs and Experimental Results

We submitted one Chinese run *TUTA1-D-C-1B* and two English runs *TUTA1-D-E-1B* and *TUTA1-D-E-2B*. For the Chinese run, the content extraction method by Qiu et al. [11] is used. For English runs, the content extraction method by Kohlschütter et al. [7] is only used for TUTA1-D-E-1B. Meanwhile, we found that: (1) there are no relevant documents in the official *qrel* file for topics 0084 and 0085. (2) No official results is provided for topic 0092. Thus, these topics are not used for our evaluation. The two initial retrieval results without taking into account the factor of diversity are used as baselines respectively. Using the metric of $D\# - nDCG@l$ [12] (l refers to the cutoff), Table 3 shows the results over all topics. Since a two-level hierarchy of subtopics is generated for each unclear query topic, *First-level*/*Second-level* refers to that the official first-level/second-level subtopics are used for evaluation.

Run	First-level ($l = 20$)	Second-level ($l = 20$)	Second-level ($l = 50$)
TUTA1-D-C-1B	0.6236	0.4598	0.4293
C-Baseline	0.3416	0.2371	0.2669
TUTA1-D-E-1B	0.5537	0.4282	0.4388
TUTA1-D-E-2B	0.3833	0.2529	0.2896
E-Baseline	0.4148	0.2978	0.3068

Table 3: Results over all topics.

From Table 3, we can observe that: (1) As the unclear topics make up the majority of all topics, selectively providing diversified result lists is important. The results also demonstrate that the diversified runs (except TUTA1-D-E-2B) outperform the baseline runs that don't take into account the factor of diversity. (2) Different from TUTA1-

[9] http://lemurproject.org/clueweb12/

D-E-1B, no specific content extraction method is deployed for TUTA1-D-E-2B. Due to noisy contents included in documents (e.g., navigational elements, templates, and advertisements), TUTA1-D-E-2B performs greatly worse that TUTA1-D-E-1B, even worse than the baseline. It is reasonable to claim that: specific content extraction helps to improve search performance. Since we mainly focus on unclear topics, Table 4 shows the results over unclear topics respectively.

Run	First-level ($l = 20$)	Second-level ($l = 20$)	Second-level ($l = 50$)
TUTA1-D-C-1B	0.6562	0.4053	0.3586
C-Baseline	0.3328	0.1728	0.2183
TUTA1-D-E-1B	0.5720	0.3916	0.4068
TUTA1-D-E-2B	0.4272	0.2398	0.2926
E-Baseline	0.4244	0.2562	0.3330

Table 4: Results over unclear topics.

The straightforward intuition about the performance over unclear topics is that: the submitted runs should perform better than they did over all topics, because the underlying models are focused on unclear topics. Comparing the results in Tables 3 and 4, we can find that: the submitted runs evaluated via the first level subtopics do perform better than they did over all topics. On the contrary, the results over the seconde level subtopics (except TUTA1-D-E-2B w.r.t. $l = 50$) are not consistent with our intuition. The possible reasons are: (1) the proposed model is not effective enough; (2) another one is the quality of the input subtopics derived from subtopic mining step.

The above experiments can demonstrate the necessity of providing diversified results for unclear topics. As for the effectiveness of the proposed model for document ranking, the comparison among different diversity models is necessary. The results summarized in the overview paper [10] show that our proposed model outperforms the others for Chinese document ranking (both coarse-grain and fine-grain evaluations) and English document ranking (fine-grain evaluation). As the runs of other teams are not provided, we didn't conduct detailed comparisons against other models.

5. CONCLUSIONS

In this paper, we described our approaches to solving the subtopic mining and document rankings subtasks in the NTCIR-11 IMine task. For the subtask of subtopic mining, our key idea is to structurally parse query-like strings by characterizing pairwise dependency on the basis of bag-of-units perspective. Then specific methods (e.g., the affinity propagation and Sainte-Laguë methods) are used to generate the target two-level hierarchy of subtopics for an unclear topic. However, the evaluation results show that the proposed approach is so effective as we though. In the future, we plan to discover the latent hierarchy among query-like strings using the deep learning technique [1].

For the subtask of document ranking, we experiment with our newly proposed 0-1 MSKP model [14]. Under this model, a subset of documents are optimally chosen like filling up multiple subtopic knapsacks. For generating the result list, we straightforwardly sorted the selected documents using their corresponding belief value in decreasing order. It would be an interesting work to explore other methods for merging the selected documents of each subtopic knapsack.

6. ACKNOWLEDGMENTS

We thank Prof. Jamie Callan and Jill Lentz for help with the ClueWeb12-B13 dataset access. Great thanks to the reviewers for helping us improve this study.

References

[1] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[2] C. L. Clarke, M. Kolla, and O. Vechtomova. An effectiveness measure for ambiguous and underspecified queries. In *Proceedings of the 2nd ICTIR*, pages 188–199, 2009.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 2009.

[4] V. Dang and W. B. Croft. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th SIGIR*, pages 65–74, 2012.

[5] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[6] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference Research on Computational Linguistics*, pages 1–15, 1997.

[7] C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the 3rd WSDM*, pages 441–450, 2010.

[8] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[9] Q. Liu and S. Li. Word similarity computing based on how-net. *Computational Linguistics and Chinese Language Processing*, 7(2):59–76, 2002.

[10] Y. Liu, R. Song, M. Zhang, Z. Dou, T. Yamamoto, M. Kato, H. Ohshima, and K. Zhou. Overview of the NTCIR-11 IMine task. In *Proceedings of NTCIR-11 Workshop*, 2014.

[11] J. Qiu, C. Tang, K. Xu, and Q. Luo. Web contents extracting for web-based learning. In *Proceedings of the 7th International Conference on Web-based Learning (ICWL 2008)*, pages 59–68, 2008.

[12] T. Sakai and R. Song. Evaluating diversified search results using per-intent graded relevance. In *Proceedings of the 34th SIGIR*, pages 1043–1052, 2011.

[13] H. Yu and F. Ren. Role-explicit query identification and intent role annotation. In *Proceedings of the 21st CIKM*, pages 1163–1172, 2012.

[14] H. Yu and F. Ren. Search result diversification via filling up multiple knapsacks. In *Proceedings of the 23rd CIKM*, accepted, 2014.