Microsoft Research Asia at the NTCIR-9 Intent Task*

Jialong Han[1,4], Qinglei Wang[2], Naoki Orii[3],
Zhicheng Dou[4], Tetsuya Sakai[4], and Ruihua Song[4]
[4]Microsoft Research Asia; [1]Renmin University of China; [2]Xi'an Jiaotong University; [3]Tokyo University
[4]{v-jihan, zhichdou, tesakai, rsong}@microsoft.com;
[2]wangqinglei0116@gmail.com; [3]orii@biz-model.t.u-tokyo.ac.jp

ABSTRACT

In NTCIR-9, we participate in the Intent task, including both the Subtopic Mining subtask and the Document Ranking subtask. In the Subtopic Mining subtask, we mine subtopics from query logs and top results of the queries, and rank them based on their relevance to the query and the similarity between them. In the Document ranking Subtask, we diversify top search results using the mined subtopics based on a general multi-dimensional diversification framework. Experimental results show that our best Chinese subtopic mining run is ranked No. 2 of all 42 runs in terms of D♯nDCG@10. Our Chinese document ranking runs generally outperform other runs in terms of I-rec. Our best Chinese document ranking runs is the No. 4 of all 24 runs in terms of D♯nDCG@10. Our Japanese document ranking runs perform the best both in terms of D-nDCG and in terms of D-nDCG.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering, Information filtering, Query formulation*

General Terms

Algorithms, Experimentation, Management, Measurement

Keywords

Query Intent, Subtopics, Diversity

NTCIR Information

Team Name: MSINT

Subtasks/Languages: Chinese Subtopic Mining, Chinese and Japanese Document Ranking

External Resources Used: Bing (just for Japanese document ranking subtask)

1. INTRODUCTION

Ever since search engines chose keywords as the interface between users and the enormous Web, the gap between user queries and their intents has more or less come into existence. With an ambiguous or broad query at hands, a search engine system may not meet the information need of the user, by simply comparing the query text with the corpus and returning a bunch of matched documents. The goal of the NTCIR9 Intent Task [9] is to understand the potential intents of a user from his vague query and use them to improve document ranking. It consists of two subtasks: Subtopic Mining and Document Ranking. In the Subtopic Mining subtask, systems are required to return a ranked list of subtopic strings in response to a given query. A subtopic could be a specific interpretation of an ambiguous query or an aspect of a faceted query. These subtopics can be used to generate diversified query suggestions or diversified results for a given query, to help users find their interested information. The document ranking subtask further explores systems to diversify search results based on mined subtopics. Systems were expected to retrieve a set of documents that covers as many intents as possible; and rank documents that are highly relevant to more popular intents higher than those that are marginally relevant to less popular intents.

We observe that intent words frequently appear along with the query words in the top results of queries. In the Subtopic Mining task, we first extract text fragments containing all query words from top query results. We then use the vector space model [6] to represent each fragment with a point in a high dimension space, and group them into clusters in purpose of discovering important sentences. Clusters are then ranked based on their relevance and importance to the query. Finally, we extract distinctive words from top clusters to generate readable subtopic strings. In addition to mining subtopics from top results, we further extract subtopics from query logs (using the SogouQ dataset) using the Maximum Result Variety (MRV) algorithm proposed by Radlinski and Dumais [5]. We further conducted multiple post-extracting processing on the subtopics mined from the two sources, including filtering, combining and ranking, to ensure that subtopics are meaningful and diversified.

In the Document Ranking task, we diversify search results using the multi-dimensional diversification framework proposed by Dou et al [4]. As the framework accepts multiple groups of subtopics, we use different types of subtopics we have mined in the Subtopic Mining task, together with the subtopics used in [4], including anchor texts, search result clusters, and website of search results.

2. SUBTOPIC MINING

2.1 Mining Subtopics from Query Logs

In the Subtopic Mining subtask, we first generate subtopics

*The work was done when the first three authors were visiting Microsoft Research Asia by analyzing the follow-up queries in sessions from query logs. We use the Maximum Result Variety (MRV) algorithm proposed by Radlinski and Dumais [5] to greedily select the set of queries that are related to the given query yet different from each other. We name the results LOG_S.

Query log-based subtopics can somehow reflect real-world user information needs, but they have the flaws as follows. Firstly, they are only available for in-log queries and may suffer from a small query log (the query log dataset for NTCIR-9 Intent Task contains only logs in one month). Secondly, they may show some bias toward background rankings. If most subtopics are already retrieved in the top results, users may just click them without issuing a new query, and hence some major subtopics cannot be found by analyzing query sessions. To solve the problem, we propose mining subtopics from top search results in the following section.

2.2 Mining Subtopics from Top Results

We observe that the phrases adjacent to the query words in top results are usually an indicator of query intents. In NTCIR-9, we propose to extract subtopics from text segments containing the original query words. Given a query q, we retrieve top K documents from a retrieval system to form a document set R. We mine subtopics from R by the following steps:

1. **Fragment Extraction** We extract several types of fragments that contain all query words from each document d in R. For the query "Mozart," "Mozart became founder of the modern Concerto," "Mozart Serenade," "Mozart effect - the more the cleverer, Amadeus is a famous movie" are some example fragments extracted.

2. **Fragment Clustering** Similar fragments are grouped to compose a cluster. For example, the fragments about Mozart's works are grouped into one cluster because they share the same item "work".

3. **Cluster Ranking** Clusters are evaluated and ranked based on how frequent their fragments occur in top results and how relevant the documents containing the fragments are. For example, the cluster on Mozart's work is ranked higher than the cluster on Mozart's birth as there are more fragments about Mozart's work extracted.

4. **Subtopic Generation** Short and readable names are generated for each cluster based on frequent phrases and associated n-grams.

2.2.1 *Fragment Extraction*

We extract the following four different types of fragments from each document d in R:

- Anchor fragment: anchor text that links to d;
- Title fragment: title text of d;
- Bold fragment: inner text of bold-like HTML tags <B> and <H1>;
- Plain fragment: sentences extracted from common HTML body text.

Note that we remove all fragments that only contain the original query as the query itself is not a subtopic. We also remove duplicate fragments in the same document because of the reason which will be introduced in Section 2.2.3.

We adopt the vector space model [6] to represent each fragment f as follows:

$$f = (w_{1,f}, w_{2,f}, ..., w_{n,f})$$

$w_{i,f}$ is the weight of a unique word i contained in f. We remove stop words and query words because they are useless to distinguish different fragments. We employ the BM25 model to calculate $w_{i,f}$, and:

$$w_{i,f} = \frac{(k_1+1)tf_i}{k_1((1-b)+b\frac{dl}{avdl})+tf_i} log \frac{N-df_i+0.5}{df_i+0.5}$$

Here tf_i is the occurrence of word i in fragment f, and df_i is the number of documents that contain i in the corpus. dl is the length of the fragment f. $avdl$ is average fragment length for the query. N is the total number of documents in the entire corpus used in NTCIR-9 (SogouT data). We experimentally set $k_1 = 1.1$ and $b = 0.7$.

2.2.2 *Fragment Clustering*

We apply a modified Partitioning Around Medoids (PAM) k-medoids algorithm clustering algorithm to group similar fragments together. The similarity between two fragments is determined using the cosine similarity between their corresponding weight vectors calculated as above. The PAM algorithm first computes k representative objects, called medoids. A medoid can be defined as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal. After finding the set of medoids, each object of the data set is assigned to the nearest medoid. k is the number of clusters we want to generate and traditionally it is fixed as an input of PAM. However, this is not suitable for our scenario. We assume one cluster represent one facet, or intent, of a query. The number of query intents is not predictable. It would be better if a clustering algorithm can decide an appropriate k. Thus, we modify the PAM algorithm as follows:

Algorithm 1 Fragment Clustering Algorithm

```
1: Randomly choose $k$ points as initial cluster medoids.
2: Assign each point $o$ to the cluster whose medoid is closest
   to $o$. If the distance between $o$ and the closest medoid
   is greater than $\theta_{threshold}$ , create a new cluster whose
   initial medoid is $o$.
3: Recalculate medoids.
4: Repeat Steps 2 and 3 until the medoids no longer change
   between two adjacent iterations.
```

2.2.3 *Cluster Ranking*

In this section, we evaluate the importance of clusters, and rank them based on importance. For this purpose, some related properties for each cluster are quantified as follows:

Document Ranking Score (DR) Intuitively, if a cluster contains the fragments that come from more and higher-ranked documents, it is more likely to correspond to one more important query intent; whereas, a cluster that contains fragments coming from only one or two documents or a single supporting website might be less significant to represent a major intent. Furthermore, different types of fragments are not equally informative. Anchor text and title of a document are usually used to summarize the topic of

Table 1: Weights of different types of fragment

Fragment Type	Weight	Fragment Type	Weight
Anchor fragment	1.0	Title fragment	0.75
Bold fragment	0.75	Plain fragment	0.5

the document or its paragraphs. In contrast with plain text fragments, they are more descriptive and are usually better sources of query intents.Base on the above assumptions, we apply the following formula to calculate document ranking score $DR(c)$ for a fragment cluster c:

$$DR(c) = \sum_{s \in sites(c)} \frac{1}{|Doc(s)|} \sum_{d \in Doc(s)} score(d, c)$$

where $sites(c)$ is the collection of websites contained in cluster c. $Doc(s)$ is the collection of documents corresponding to the domain s in c, and $score(d, c)$ is the overall score of all fragments extracted from d, and we let

$$score(d, c) = \frac{\max_{f \in Frag(d,c)} w(f)}{\sqrt{Rank(d)}}$$

where $Frag(d, c)$ is the set of fragments in C extracted from d. $w(f)$ is a weight determined by the type of the fragment. Table 1 shows the weight settings of each fragment type.

Inverted Average Length (IAL) We denote the count of words contained in a fragment f with $Len(f)$. Generally, a shorter name is preferred for intent representation, and hence we think that a shorter fragment is a better source of subtopics. Suppose $Frag(c)$ is the collection of fragments in cluster c, we use the following equation, which we refer to as Inverted Average Length (IAL), to measure the importance of a cluster in terms of average length of the fragments.

$$IAL(c) = \frac{|Frag(c)|}{\sum_{f \in Frag(c)} Len(f)}$$

Punish Score (PS) We observed ten example Chinese topics and found some situations where a cluster is not a good candidate and should be punished. For instance, clusters containing fragments of few types or from few websites are less informative to represent an important query intent. We publish these kinds of clusters by a punished score (PS) defined as below:

$$PS(c) = -\frac{1}{|Type(c)| * |sites(c)|}$$

where $|Type(c)|$ is the number of fragment types in cluster c.

Given the above three features, we simply use a linear combination to calculate a single score for each cluster as follows.

$$Score(c) = w_1 DR(c) + w_2 IAL(c) + w_3 PS(c)$$

We empirically set $w_1 = 0.7$, $w_2 = 0.2$, and $w_3 = 0.1$ in this paper.

2.2.4 Generating subtopics for clusters

In this section, we generate a readable description from a cluster as a candidate subtopic. We first select the most frequent word and extend it to an n-gram. Then, we find the shortest string containing words of the n-gram and the given query as a candidate subtopic. The stop words are counted in n-gram generation, so that they could be shown when they are adjacent to meaningful keywords in the subtopic. The process is repeated until the candidate subtopic is different from existing candidate subtopics. The algorithm can be summarized as follows:

Algorithm 2 Subtopic Generation

```
Require:
    Original query $q$
    Word set $W = \Phi$
    N-gram set $NG = \Phi$
    Subtopic set $S = \Phi$
for each $c \in C(q)$ do
    $//C(q)$ is all fragment clusters for query $q$
    repeat
        Select most frequent word $w$
        Extend $w$ to n-gram ng
    until $w \notin W \& ng \notin NG$
    $W = W \cup \{w\}$
    $NG = NG \cup \{ng\}$
    for each $f \in F(c)$ do
        $//F(c)$ is fragment collection for cluster $c$
        Find the shortest string $s$ as candidate subtopic
        S.t. $W_s = W_q \cup W_{nq} \cup \{$other words$\}$
    end for
    Find the most frequent subtopic $s_{fre}$
    $S = S \cup \{s_{fre}\}$
end for
return $S$
```

2.3 Subtopic Ranking and Diversification

As we rank clusters and generate subtopics from clusters by order, the extracted subtopics are naturally ranked in terms of the importance of corresponding clusters. We call the ranked subtopics DOC_S. With LOG_S and DOC_S at hands, we propose three post-processing methods to combine them, rank them, and filter low-quality subtopics. Our final runs are based on the combination of the two subtopic sources and three post-processing methods.

2.3.1 Language-Model-Based Reranking

The first method, LM, is to measure the importance of subtopics from users' perspective. We propose using SogouQ to learn a language model [10]. For extracted subtopic list S, we first calculate a score for each subtopic. The language model is then applied to rank S based on the score. We call the ranked subtopics LM(S). In the language model, we assume each word depends only on the previous four words.

2.3.2 SVM-Based Filtration

Another method is to filter out low-quality subtopics from a given subtopic list S. We call the new subtopic list Filter(S). We observe DOC_S and find some of them are not complete or are long sentences to describe a fact about the given query, rather than a subtopic or facet of the query. We argue that such kinds of subtopics are of low-quality and would be removed from the list of ranked subtopics. Thus, we formulate the problem of identifying noisy subtopics as a classification problem. First, we labeled noisy subtopics among top 30 extracted subtopics for ten Chinese example topics. Then, we extract 16 features including the score calculated by language model and 15 POS (part-of-speech) fea-

Table 2: 15 POS features used in the SVM training

	NT	VV	DEC	AD	CD
Number of	M	VA	LC	DT	VE
	DEG	P	CC	OD	BA

tures, e.g., the number of terms tagged as NT, the number of VVs, and the number of DECs (see Table 2[1]). We apply Stanford POS tagger [11] in our experiments. Next, we adopt Support Vector Machines (SVMs) developed by Vapnik [1] with RBF (Radial Basis Function) kernel to train a classifier to identify noisy subtopics. When we conduct tenfold cross validation experiments upon the training dataset, the best classifier in our experiments achieves a precision of 74.7% and a recall of 87%. This indicates that noisy subtopics we defined can be automatically identified. Finally, we apply the classifier learned from the ten example topics to filter noisy subtopics for 100 formal Chinese topics in our formal run.

2.3.3 MMR-based Reranking

We utilize the well-known MMR [2] (short for Maximum Marginal Relevance) framework to further evaluate the diversity of mined subtopics. The MMR model treats the ranking problem as a procedure of successively selecting the "best" unranked object (usually a document but in our scenario, a subtopic) and arranging it at the tail of the rank list. Unlike non-diversified ranking where the quality of an object is simply its degree of relevance, the MMR model makes a compromise between relevance and diversity: when looking for the next best object, it chooses not the most relevant one, but the one that is both relevant and novel (not resembling those objects that are already chosen and ranked).

Given the relevance function $Rel(.)$ and similarity function $Sim(.,.)$, the MMR model could be set up as following:

$$q_{i+1} = \arg\max_{q \notin Q_i} \{\alpha Rel(q) + (1-\alpha) Nov(q, Q_i)\}$$

where α in $[0,1]$ is a combining parameter, and then

$$Q_{i+1} = Q_i \cup \{q_{i+1}\}$$

Where q_i is the object ranked at the i-th position and Q_i is the collection containing the top-i objects. The function $Nov(q, Q_i)$ tries to characterize the novelty of object q in the case that Q_i is already chosen. In practice, the novelty function could be implemented using the similarity function $Sim(.,.)$. The following is an example, whose intuition is minimizing the similarity between the current object and its most resembled ranked object:

$$Nov(q, Q_i) = -\max_{q' \in Q_i} Sim(q, q')$$

As for the implementation of the relevance and similarity function for subtopics, it is often a harder problem than that for the documents. The reasons lie in the form of subtopics - they are often short, carrying little information. Therefore we consider expanding the subtopics with the documents that is relevant to it. In practice, we retrieve the top relevant documents $\{d_j\}$ together with their rank scores for each subtopic q. For any two subtopics q_1 and q_2, if they share many top-ranked documents and put high scores on them, they tend to be on the same topic. Denoting the retrieved documents of q_1 and q_2 as $\{d_j\}^1$ and $\{d_j\}^2$ respectively, we use the following equation to model our intuition towards document set similarity:

$$DocSim(q_1, q_2) = \sum_{d \in \{d_j\}^1 \cap \{d_j\}^2} \sqrt{score_1(d) * score_2(d)}$$

Another signal that indicates subtopic similarity is their string similarity. We use

$$StrSim(q_1, q_2) = JaccardSim(q_1, q_2) + 1/EditDist(q_1, q_2)$$

as an implementation. To obtain a unified similarity function, we linearly combine the document set similarity and string similarity function as follows, where β is the combination parameter:

$$Sim(q_1, q_2) = \beta StrSim(q_1, q_2) + (1-\beta) DocSim(q_1, q_2)$$

So far we have derived the subtopic similarity function Sim(.,.). As for the subtopic relevance function Rel(.), we notice the fact that an original query q_0 always has nothing different in the form from its subtopics mined in the previous phase. They're all in the form of short queries, whose intent could be found both from their text content and their relevant documents. So we simply define the relevance of a subtopic as the similarity between the subtopic and its original query, that is,

$$Rel(q) = Sim(q_0, q)$$

2.4 Submitted Runs

We submit the following five runs for the Chinese Subtopic Mining subtask:

- MSINT-S-C-1: combine LOG_S, Filter(DOC_S) and LM(DOC_S), and diversify the merged list by both string similarity and search results similarity(setting $\beta = 0.7$). No external resource is used.
- MSINT-S-C-2: combine LOG_S, Filter(DOC_S) and LM(DOC_S), and diversify the merged list only by string similarity (setting $\beta = 1$). No external resource is used.
- MSINT-S-C-3: Filter(DOC_S), extract subtopics from documents and filter them by the classifier. No external resource is used.
- MSINT-S-C-4: LM(DOC_S), extract subtopics from documents and rank them by the language model. No external resource is used.
- MSINT-S-C-5: DOC_S, extract subtopics from documents and rank them based on the importance of clusters. No external resource is used.

3. DOCUMENT RANKING

3.1 Search Result Diversification Framework

For the Document Ranking submissions, we built upon the multi-dimensional diversification framework proposed by

[1] See http://www.cis.upenn.edu/~chinese/posguide.3rd.ch.pdf for a detailed explanation of the POS terms

Dou et al. [3, 4]. For a given ambiguous query q, we first create an initial document ranking R, using Microsoft's internal web search platform WebStudio [2]. A greedy algorithm then iteratively selects the best document from R and creates a diversified ranking S_n, using the following equation to choose the document at each step:

$$d_{n+1} = \underset{d \in R \setminus S_n}{\arg\max} [\rho \cdot r(q, d) + (1 - \rho) \cdot \Phi(d, S_n, \mathbb{C})]$$

Here, $r(q, d)$ represents the relevance of document d with respect to q, and $\Phi(d, S_n, \mathbb{C})$ represents the diversity of d with respect to dimensions $\mathbb{C}$, with already-selected documents S_n taken into consideration. ρ is a parameter that adjusts the tradeoff between relevance and diversity. Dimensions refer to different aspects of the ambiguity present in q, and are taken from different data sources (i.e. anchor texts, query logs, clusters of search results, etc).

3.2 Chinese Runs

For the Chinese run submissions, search results are diversified based on websites of top results, anchor texts, and subtopics that are mined in the Subtopic Mining task described in the previous sections. Details of the first two types of subtopics can be found in [3, 4]. We submit five runs with the following configurations of subtopics:

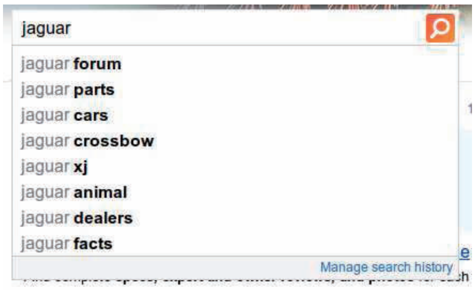
- MSINT-D-C-1: diversify top results based on five types of subtopics: websites of top results, anchor texts, LOG_S, Filter(DOC_S), and LM(DOC_S)
- MSINT-D-C-2: diversify top results based on three different types of subtopics: websites of top results, anchor texts, and combined subtopics MSINT-S-C-1
- MSINT-D-C-3: the baseline ranking without any diversification
- MSINT-D-C-4: diversify top results based on four types of subtopics: websites of top results, anchor texts, LOG_S, and Filter(DOC_S)
- MSINT-D-C-5: diversify top results based on four types of subtopics: websites of top results, anchor texts, LOG_S, and LM(DOC_S)

3.3 Japanese Runs

For the Japanese runs, we use the following subtopic dimensions: suggested queries shown by web search engines (WSEs), related queries shown by WSEs, and domain names of top results. Our run submissions consist of different combination of these three dimensions.

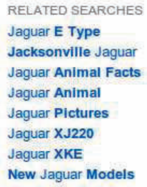
Query reformulations provided by WSEs have been shown to effectively diversify search results [7]. In this task, we utilize two types of query reformulations provided by Bing[3]:

- suggested queries: variants of the original query shown in a drop-down list as the user types the query into the search box. Figure 1 shows an example of suggested queries. As shown in the caption for the figure, note that we append a space after the query 'jaguar', for the list differs when with or without a space.



jaguar
jaguar forum
jaguar parts
jaguar cars
jaguar crossbow
jaguar xj
jaguar animal
jaguar dealers
jaguar facts
Manage search history

Figure 1: Suggested queries for the query 'jaguar'



RELATED SEARCHES
Jaguar E Type
Jacksonville Jaguar
Jaguar Animal Facts
Jaguar Animal
Jaguar Pictures
Jaguar XJ220
Jaguar XKE
New Jaguar Models

Figure 2: Related queries for the query 'jaguar'

- related queries: variants of the original query shown in the search result page, typically alongside the list of documents. An example of related queries is shown in figure 2.

Both types of query reformulation were extracted in the middle of June 2011.

Our run configurations for the submission are as follows:

- MSINT-D-J-1: diversify top results using suggested queries, related queries, and websites
- MSINT-D-J-2: diversify top results using suggested queries and websites
- MSINT-D-J-3: diversify top results using suggested queries
- MSINT-D-J-4: diversify top results using related queries and websites
- MSINT-D-J-5: baseline (no diversification performed)

In all of our runs, we empirically set the relevance-diversity tradeoff parameter ρ as 0.6, but selective diversification using query-dependent ρ is a viable option [8].

4. EXPERIMENTAL RESULTS

4.1 Chinese Subtopic Mining Runs

Table 3 shows the evaluation results of our submitted subtopics. We observe that the run MSINT-S-C-2 combining multiple post-processing methods outperforms the runs involving only one method (MSINT-S-C-3, MSINT-S-C-4, and MSINT-S-C-5). Although there is a slight change in the order between MSINT-S-C-2 and MSINT-S-C-5 from cut-off at 20 to 30, the overall trend is similar for all three cut-off levels: MSINT-S-C-2 attains best performance throughout all levels, followed by MSINT-S-C-4, MSINT-S-C-5, and MSINT-S-C-3. Applying language model to rank subtopics

[2] WebStudio platform: `http://research.microsoft.com/en-us/projects/webstudio/`

[3] Bing, `http://www.bing.com`

Table 3: Chinese Subtopic Mining runs ranked by D♯-nDCG at various cut-off levels

cut-off	run name	I-rec	D-nDCG	D♯-nDCG
@10	MSINT-S-C-2	0.513	0.6806	0.5968
	MSINT-S-C-4	0.4864	0.6604	0.5734
	MSINT-S-C-1	0.5002	0.624	0.5621
	MSINT-S-C-5	0.4578	0.6543	0.556
	MSINT-S-C-3	0.4587	0.6256	0.5422
@20	MSINT-S-C-2	0.6066	0.6462	0.6264
	MSINT-S-C-4	0.6293	0.6008	0.615
	MSINT-S-C-5	0.6069	0.6122	0.6096
	MSINT-S-C-3	0.5962	0.5852	0.5907
	MSINT-S-C-1	0.6187	0.5506	0.5846
@30	MSINT-S-C-5	0.65	0.5412	0.5956
	MSINT-S-C-4	0.6638	0.515	0.5894
	MSINT-S-C-2	0.6275	0.539	0.5832
	MSINT-S-C-3	0.6218	0.5022	0.562
	MSINT-S-C-1	0.6432	0.4662	0.5547

Table 4: Results of Chinese document ranking runs. Note that MSINT-D-C-3 is the baseline ranking without diversification.

cut-off	run name	I-rec	D-nDCG	D♯-nDCG
@10	MSINT-D-C-3	0.5987	0.3222	0.4604
	MSINT-D-C-1	0.7068	**0.3854**	**0.5461**
	MSINT-D-C-2	0.7003	0.3783	0.5393
	MSINT-D-C-4	**0.7091**	0.3822	0.5456
	MSINT-D-C-5	0.6936	0.3783	0.5359
@20	MSINT-D-C-3	0.7245	0.3304	0.5274
	MSINT-D-C-1	0.8055	**0.3836**	0.5946
	MSINT-D-C-2	0.801	0.3828	0.5919
	MSINT-D-C-4	0.8013	0.3806	0.5909
	MSINT-D-C-5	**0.8095**	0.3801	**0.5948**
@30	MSINT-D-C-3	0.7719	0.3156	0.5437
	MSINT-D-C-1	0.8343	**0.3645**	**0.5994**
	MSINT-D-C-2	0.8327	0.3625	0.5976
	MSINT-D-C-4	**0.8349**	0.3609	0.5979
	MSINT-D-C-5	0.833	0.3619	0.5974

is a little better than ranking subtopics based on the importance of belonging clusters. However, DOC_S filtered by the classifier is a little worse. Comparing MSINT-S-C-2 and MSINT-S-C-1, we see that most of the time the diversity considering search results similarity has a negative effect on D♯-nDCG. Maybe it's due to the noise in the documents that blurs the similarity function from distinguishing different intents in the subtopics.

4.2 Chinese Document Ranking Runs

Experimental results of our submitted Chinese document ranking runs are shown in Table 4. We find that:

(1) The runs with diversification (MSINT-D-C-1, MSINT-D-C-2, MSINT-D-C-4, and MSINT-D-C-5) outperform the baseline run (MSINT-D-C-3), no matter in terms of I-rec, D-nDCG, or D♯-nDCG. This indicates that the diversification framework is effective in improving diversity of search results.

(2) Comparing MSINT-D-C-1 and MSINT-D-C-2, we found that using the original three types of subtopics LOG_S, Filter(DOC_S), and LM(DOC_S) is always better than using the merged and diversified subtopics MSINT-S-C-1. This means that when a multi-dimensional diversification framework is used, we can fully utilize the information contained in different types of subtopics. A relevant subtopic may duplicate in multiple data sources, and hence its corresponding documents would be ranked higher.

(3) MSINT-D-C-1 performs the best result in terms of D-nDCG and D♯-nDCG, which indicates that using more data sources can help improve document diversity.

4.3 Japanese Document Ranking Runs

Table 5 shows the results of our submitted runs for INTENT at various cut-off levels. We observe that all of the diversified runs (MSINT-D-J-1 to MSINT-D-J-4) improves upon the baseline (MSINT-D-J-5). Although there is a slight change in the order between MSINT-D-J-2 and MSINT-D-J-1 from cut-off at 10 to 20, the overall trend is similar for all 3 cut-off levels: MSINT-D-J-3 attained best performance throughout all levels, followed by MSINT-D-J-1/MSINT-D-J-2, MSINT-D-J-4, and finally the baseline.

Comparing MSINT-D-J-5 and MSINT-D-J-4, we see that enabling related queries and website dimensions have a positive effect on D♯-nDCG. However, comparing MSINT-D-J-1 and MSINT-D-J-3, we see that disabling related queries and websites, while keeping suggested queries enabled, also has a positive effect. Thus, we see that dimensions are not independent of each other. How to determine the optimal choice of dimensions, and how to determine the optimal weighting scheme for these dimensions have room for further investigation.

5. CONCLUSIONS

In this paper, we introduce our approaches and system for participating in the NTCIR9 Intent Task. Most of our work concentrates on how to mine subtopic strings from the documents, in order to make up for a small and sparse query log. Results show that our methods are practically feasible. After the contest results are released, while we do not involve any external data resource, we are pleased to see that other teams made good use of resources from Wikipedia and "Baidu Knows" (the most popular Wikipedia in China) and achieved good performances in their submissions. We believe that this reflects a fact: when building an application to meet the subtle and various demands of human beings, rather than designing too complicated models, the easiest way out is to utilize the nutritious dataset generated by human itself - the bigger, the more real, the better.

6. REFERENCES

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[2] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents

Table 5: Japanese Document Ranking runs ranked by D♯-nDCG at various cut-off levels. Note that MSINT-D-J-5 is the baseline ranking without diversification

cut-off	run name	I-rec	D-nDCG	D♯-nDCG
@10	MSINT-D-J-3	0.7554	0.4444	0.5999
	MSINT-D-J-2	0.7548	0.4329	0.5938
	MSINT-D-J-1	0.7488	0.4377	0.5933
	MSINT-D-J-4	0.7526	0.4210	0.5868
	MSINT-D-J-5	0.7063	0.4257	0.5660
@20	MSINT-D-J-3	0.8697	0.4821	0.6759
	MSINT-D-J-1	0.8669	0.4644	0.6656
	MSINT-D-J-2	0.8720	0.4577	0.6648
	MSINT-D-J-4	0.8563	0.4519	0.6541
	MSINT-D-J-5	0.8372	0.4692	0.6532
@30	MSINT-D-J-3	0.9058	0.4813	0.6936
	MSINT-D-J-1	0.9025	0.4678	0.6852
	MSINT-D-J-2	0.8968	0.4464	0.6716
	MSINT-D-J-4	0.8850	0.4383	0.6617
	MSINT-D-J-5	0.8679	0.4486	0.6583

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] Z. Dou, K. Chen, R. Song, Y. Ma, S. Shi, and J.-R. Wen. Microsoft research asia at the web track of trec 2009. In *Proceedings of TREC 2009*, 2009.

[4] Z. Dou, S. Hu, K. Chen, R. Song, and J.-R. Wen. Multi-dimensional search result diversification. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 475–484, New York, NY, USA, 2011. ACM.

[5] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 691–692, New York, NY, USA, 2006. ACM.

[6] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, November 1975.

[7] R. L. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 881–890, New York, NY, USA, 2010. ACM.

[8] R. L. Santos, C. Macdonald, and I. Ounis. Selectively diversifying web search results. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1179–1188, New York, NY, USA, 2010. ACM.

[9] R. Song, M. Zhang, T. Sakai, M. Kato, Y. Liu, M. Sugimoto, Q. Wang, and N. Orii. Overview of the ntcir-9 intent task. In *Proceedings of NTCIR-9*, 2011.

[10] A. Stolcke. Srilm - an extensible language modeling toolkit, 2002.

[11] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.