Multi-Aspect Query Summarization by Composite Query*

Wei Song[1], Qing Yu[2], Zhiheng Xu[3], Ting Liu[1], Sheng Li[1], Ji-Rong Wen[2]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China
{wsong, tliu, lisheng}@ir.hit.edu.cn
[2] Microsoft Research Asia, Beijing, 100190, China
{qingyu, jrwen}@microsoft.com
[3] Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
xuzhiheng19881130@gmail.com

ABSTRACT

Conventional search engines usually return a ranked list of web pages in response to a query. Users have to visit several pages to locate the relevant parts. A promising future search scenario should involve: (1) understanding user intents; (2) providing relevant *information* directly to satisfy searchers' needs, as opposed to relevant *pages*. In this paper, we present a paradigm for dealing with informational queries. We aim to summarize a query's information from different aspects. Query aspects are aligned to user intents. The generated summaries for query aspects are expected to be both specific and informative, so that users can easily and quickly find relevant information. Specifically, we use a *"Composite Query for Summarization"* method, which leverages the search engine to proactively gather information by submitting multiple composite queries according to the original query and its aspects. In this way, we could get more relevant information for each query aspect and roughly classify information. By comparative mining the search results of different composite queries, it is able to identify query (dependent) aspect words, which help to generate more specific and informative summaries. The experimental results on two data sets, Wikipedia and TREC ClueWeb2009, are encouraging. Our method outperforms two baseline methods on generating informative summaries.

Categories and Subject Descriptors

H.3.m [**Information Storage and Retrieval**]: Miscellaneous

General Terms

Algorithms, Experimentation

*This work was done when the first and third authors were visiting Microsoft Research Asia

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

Keywords

Query aspect, Query summarization, Composite query, Mixture Model

1. INTRODUCTION

Nowadays, accessing information on the Internet through search engines has become a fundamental life activity. Current web search engines usually provide a ranked list of URLs to answer a query. This type of information access does a good job for dealing with simple navigational queries by leading users to specific websites. However, it is becoming increasingly insufficient for queries with vague or complex information need. Many queries serve just as the start of an exploration of related information space. Users may want to know about a topic from multiple aspects. Organizing the web content relevant to a query according to user intents would benefit user exploration. In addition, a list of URLs couldn't directly satisfy user information need. Users have to visit many pages and try to find relevant parts within long pages, since the information may be scattered across documents. The long-standing goal of search engines should be providing relevant *information*, as opposed to relevant *documents*, to directly satisfy searchers' needs.

This paper presents a novel search paradigm that the system should automatically discover information and present an informative overview for a query from multiple aspects. We target on dealing with informational queries. A query represents a centric topic, and the query aspects are aligned to user intents covering diverse information needs. The query aspects could be specified explicitly by users through an interface or automatically mined from search logs or other resources [4, 18, 22, 25]. In this paper, we use simple methods to do aspect mining and mainly focus on ***multi-aspect oriented query summarization***: given a query and a set of aspects, generate a summary for each query aspect, which is expected to provide specific and informative content to users directly and helps for further exploration. Figure 1 shows an example of the system output.

We further formulate the multi-aspect oriented query summarization into 2 phases: *information gathering* and *summary generation*. Different from traditional text summarization where a set of documents to be summarized is given as a system input, we propose a *"Composite Query for Summarization"* method, which leverages the search engine to proactively gather related information. In addition to using the search result of the original query, we also composite a set of new queries and submit them to the search engine to

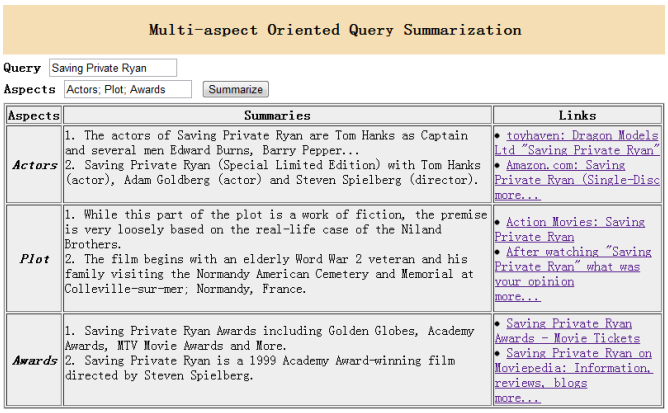

Figure 1: An example output of multi-aspect oriented query summarization.

collect query aspect related information. For example, by concatenating the original query and the keywords of an aspect as a query, we are able to get query dependent aspect information; by submitting the aspect keywords only as a query, we could get query independent aspect information. Our motivations are:

First, the search result of the original query may not contain enough information for all aspects that users care about, because the search engine returns documents only considering whether a document is relevant to the query keywords, rather than its aspects.

Second, for better aspect oriented exploration, the information for different query aspects should be as orthogonal as possible. It is important to distinguish the aspect specific information from the general information about the whole query. By using the composite queries, we could get more specific information for each aspect.

The flexible information gathering also helps for *summary generation* phase. By comparing the search results of different types of composite queries, query (dependent) aspect words can be identified without complex natural language processing, based on which more specific and informative summaries could be generated,

The contributions of this paper can be summarized as follows:

- We formulate the multi-aspect based query summarization task. In this scenario, the system proactively discovers information and aims to provide *multiple dimensional* and *direct* information seeking in response to informational queries.

- We propose a *"Composite Query for Summarization"* method for proactive information gathering, which is a key point for our task and differs from traditional search result organization and textual summarization.

- We emphasize generating specific and informative summaries to directly address searchers' needs on different aspects. To achieve this, we propose a simple method to identify query aspect dependent words by comparing the search results of different types of composite queries.

- We conduct experiments on both real web queries and large-scale pseudo queries based on Wikipedia[1]. Automatic evaluation and human judgements are used for measuring the quality of generated summaries.

The rest of the paper is organized as follows. First, we discuss related work in Section 2. In Section 3, we define the query aspect and briefly introduce optional approaches for query aspect mining. In Section 4, we detail the proposed "composite query" based method for both information gathering and aspect oriented summary generation. After that, we report our experimental results in Section 5. Section 6 states our conclusions.

2. RELATED WORK

2.1 Search Result Organization

Exploratory search becomes a new frontier in the search domain, which aims to provide additional support for information seeking beyond simple lookup [24]. Recent work has shown that well-organized search results are helpful for information exploration. For example, search result clustering [9, 11, 27], categorizing [1], facet based information exploration [6], representative queries [23] and tag clouds [10] are adopted for search result navigation. Clustering based approaches automatically group similar search result documents together [9, 11, 27]. Search result documents can also be classified into a manually constructed category taxonomy [1]. But the fixed hierarchy often lacks of flexibility to describe various user information needs. Faceted search aims to offer the ability for searchers to filter search results by specifying desired attributes [6]. However, the facets are usually pre-defined for some specific domain so that it is difficult to apply it to web search. Though most of the above methods organize search result documents into various aspects and improve user experience for information exploration, the content are still presented at document level, and users can't get relevant information directly.

2.2 Document Summarization

Single document summarization techniques have been successfully applied in web search engines (snippet generation) [19, 20]. A span of text gives users a first sight of the topics of a document. For efficiency, sentence extraction strategy is used for generating query dependent summaries [5].

Comparing with single document summarization, multi-document summarization is expected to generate a global picture for a set of documents which is given as input [15, 26]. Recently researchers utilize latent topics for multiple document summarization. For example, subtopics from the narrative of a topic (a description of a topic, which is provided by the DUC summarization track) is used to enhance summarization [17]. Wang uses topic model to extract subtopics and select sentences by topic words [21]. However, the latent topics used in these papers are usually mined unsupervised. As a result, the topics may fit to the data collection, rather than align to user intents.

Some work makes use of predefined aspects to provide (sentiment) summarization on reviews or comments [7, 14]. Our work is also inspired by [13], which incorporates user interaction into the summarization process. Given a corpus of documents, users predefine their interested facets and the

[1]http://en.wikipedia.org/

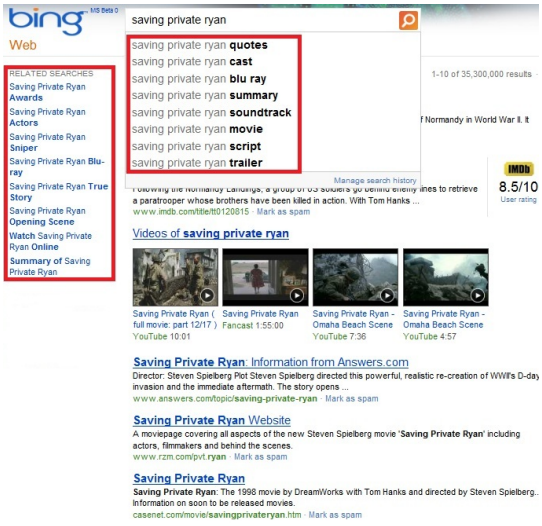

Figure 2: A snipping of returned documents for query "Saving Private Ryan" and two typical services provided by Bing Search.

system provides summaries according to the facets. The authors evaluate it on online reviews and Gene corpus (which are relatively "clean" data sets). In contrast, we focus on summarizing user intents related to a query rather than a given corpus. They don't consider the informativeness of the generated summaries, while one of our goals is to provide direct information to users.

Our work is based on query aspects but differs from existing work in several points. First, in our framework, query aspects could be mined from any resources but not limited to a set of documents to be summarized. Second, the traditional summarization task treats the documents as a given input to the system. However, in our scenario, we separate the information gathering and summarization generation phases. In this way, we view the whole web as a corpus and could proactively collect more related information for summarization. Third, we aim to generate both specific and informative content for each query aspect. Therefore, users could get relevant information directly.

3. QUERY ASPECT

Multi-aspect oriented query summarization depends on query aspects. In this section, we define the query aspect and briefly discuss query aspect mining methods both in literature and in realistic way.

An aspect represents a distinct information need relevant to the original query. Recently, various methods have been proposed for automatically discovering query intents [2, 4, 22, 25]. The NTCIR-9 Intent Task was organized to explore and evaluate the technologies of mining and satisfying different user intents for a vague query [18]. In these work, a query aspect is represented in different ways, such as a set of search queries related to the original query [2, 25], a set of query qualifiers [22] or a single intent string [18]. These definitions are in fact very similar. The main differences are: (1) Whether distinguish the original query and the query qualifier. (2) Whether select an exemplar (label) to represent a set of queries related to the same intent.

Inspired by previous work, we define an **aspect** as a query qualifier - keywords that are added to an **original query** to form a specific user intent. For example, "reviews" and "actors" could be seen as aspects for a movie. In this work, we mainly focus on multi-aspect oriented summary generation and use very simple method to mine query aspects. However, any existing method for mining query aspects could be incorporated.We can also use the services provided by search engines to get approximate query aspects. For example, search engines provide "query suggestion" or "related searches" features. Figure 2 shows a snipping of the search result from Bing Search page for query "Saving Private Ryan", a famous movie. Thus, the aspects could be easily identified using simple rules from related searches. We could also predefine some aspect templates for certain query classes, such as movie, travel, music, people, etc. We leave this as future work.

4. MULTI-ASPECT ORIENTED QUERY SUMMARIZATION

Now, we suppose the aspects are given and aim to summarize a query according to its different aspects. We expect to generate both *specific* and *informative* summary for each aspect instead of a set of documents so that the users could get relevant information directly. First, we explain the meaning of **specific** and **informative** by an example. Suppose that for the query "Saving Private Ryan", one of the user information needs is to know the "actors" of this movie. There are some candidate sentences:

(i) *"A movie page covers information about new Steven Spielberg movie 'Saving Private Ryan' including actors, film makers and behind the scenes."*

(ii) *"Saving Private Ryan cast are listed here including the Saving Private Ryan actresses and actors featured in the film."*

(iii) *"The actors of Saving Private Ryan are Tom Hanks as Captain and several men Edward Burns, Barry Pepper..."*

All the three sentences contain certain information about the aspect "actors". The first one talks about the general information about the query. It is Not specific to the desired aspect. The second sentence focuses on the desired aspect, however, it does not provide relevant information directly, only gives navigational information. We say it is specific but Not informative. The third sentence should be a good candidate which provides direct answers to the desired aspect, i.e., the names of the actors. It is both specific and informative.

As the example shows, the challenges of this task include: (1) Distinguish aspect specific information from general query information. (2) Identify informative content instead of navigational information only. We take the *Composite Query for Summarization* method to deal with above issue, which consists of 2 phases: information gathering and summary generation. First, we proactively get aspect specific information using composite queries. Then a mixture model is used to model different types of words which present query common information or aspect specific information. Finally, we rank the candidate sentences based on the mixture model and the redundancy in search results for generating summaries.

4.1 Information Gathering

Existing work on text summarization doesn't pay much attention on how to collect data. A natural way is to use query search result. However, there may be not enough information for certain query aspects, if we only use the search result of the original query. For example, some users wonder whether movie "Saving Private Ryan" tells a true story, but few top documents in the search result of "Saving Private Ryan" discuss this topic.

We present a composite query based method for information gathering. Formally, we denote the original query as Q and an aspect as A_k. For example, Q refers to the original query "Saving Private Ryan" and A_k refers to one aspect "actors". In information gathering phase, we composite a new query by concatenating the original query and the aspect words, denoted as $Q + A_k$. The composite query is "Saving Private Ryan actors". Therefore, we can submit the composite query to the search engine to get top ranked documents. Comparing with the search result of the original query, the search result of the composite query is much more specific for the query aspect. Also, we can submit the aspect A_k itself to the search engine to get information about the aspect which is query independent.

For a query with K aspects, we have a set of composite queries $\{Q, Q + A_1, ..., Q + A_K, A_1, ..., A_K\}$. We use the top returned documents for each composite query. The search result of Q (denoted as C_Q) provides overall information about the query; the search result of $Q + A_k$ (denoted as C_{Q+A_k}) provides the information about the aspect A_k of the query Q. The search result of A_k (denoted as C_{A_k}) provides information about the aspect itself which is query independent. The idea of using composite queries is straightforward and the benefits are two folded: (1) We collect more aspect related data which may be not contained in original query's search result. (2) The search engine helps us roughly classify information according to the query aspects.

Based on the collected data for query aspects, we identify aspect words by comparing the search results of different types of composite queries. These words are then used for assisting summary generation.

4.2 Summary Generation

4.2.1 Modeling Search Result

We assume the desired information for query aspect A_k is embedded in collection C_{Q+A_k}, which consists of 3 kinds of information: query general information, aspect information, irrelevant information. Correspondingly, the words in search results could be divided into 3 categories:

Query Common Words: They tend to occur frequently across multiple aspects, such as "movie", "TV", "IMDB" for "Saving Private Ryan".

Query Aspect Words: These words provide information for an aspect, such as "cast", "list" and "Tom Hanks" for the aspect "actors".

Global Background Words: These words distribute heavily on the Web. Mostly, they are stop words or high frequency non-discriminative words.

Figure 3 shows 3 types of words and their relationship in search results of the original query and the composite queries. We assume that the query aspect words describing the aspect A_k of query Q will occur more in C_{Q+A_k}, while the query common words will occur frequently across multiple aspects. Based on the collected data by using composite queries, the observations support the assumption. Therefore, we adopt a mixture model to describe each type of words. Formally, θ_k represents the *query aspect words* model for aspect A_k. θ_B represents the *query common words* model. θ_G represents the *global background words* model which is to draw globally high frequency terms. All these models are multinomial probability distributions over vocabulary.

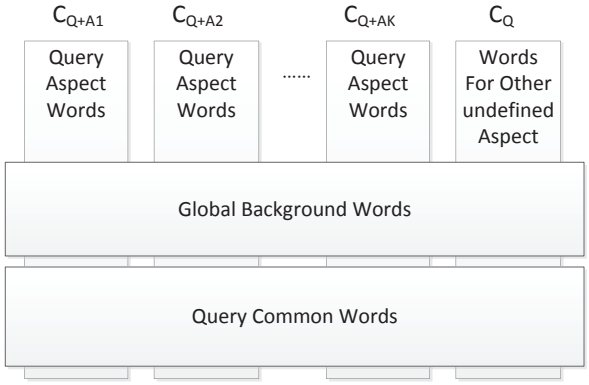

Figure 3: The illustration of the relationship between the search results of different composite queries and different types of words.

The collection C_{Q+A_k} could be generated by the mixture model. Each word w in C_{Q+A_k} is generated according to:

$$p_k(w) = \lambda_G p(w|\theta_G) + (1 - \lambda_G) \times (\alpha_B p(w|\theta_B) + (1 - \alpha_B) p(w|\theta_k)) \quad (1)$$

where $p_k(w)$ represents the probability of a term occurrence w in collection C_{Q+A_k}, λ_G and α_B are fixed parameters. The generative process could be seen as 2 steps: first decide whether this word is from θ_G, and then decide it comes from θ_B or θ_k. To estimate the aspect word model θ_k, we first estimate θ_G and θ_B. θ_G is estimated using maximum likelihood estimator based on document frequency which is computed on a large collection of web pages. θ_B is estimated by combining the search results of the original query and all query aspects, i.e., $C_Q \cup \{C_{Q+A_k}\}$. We use C_Q to catch the general content of the query and the unknown aspects which are not defined explicitly or mined already. θ_B could be estimated according to:

$$p(w|\theta_B) = \frac{tf(w, C_Q) + \sum_k tf(w, C_{Q+A_k})}{\sum_{w'} (tf(w', C_Q) + \sum_k tf(w', C_{Q+A_k}))} \quad (2)$$

where $tf(w, \cdot)$ represents the term frequency in a collection. After deriving $p(w|\theta_B)$ and $p(w|\theta_G)$, $p(w|\theta_k)$ could be estimated using the expectation maximization (EM) algorithm [3] by maximizing the log-likelihood of the collection C_{Q+A_k}:

$$L(C_{Q+A_k}) = \sum_w \log tf(w, C_{Q+A_k}) p_k(w) \quad (3)$$

For each term w in C_{Q+A_k}, the updating formulas of the E-step and the M-step are shown below:

E-Step:

$$p_w(z=G) = \frac{\lambda p(w|\theta_G)}{\lambda p(w|\theta_G)+(1-\lambda)((1-\alpha_B)p(w|\theta_k)+\alpha_B p(w|\theta_B))}$$

$$p_w(z=k) = \frac{(1-\alpha_B)p(w|\theta_k)}{(1-\alpha_B)p(w|\theta_k)+\alpha_B p(w|\theta_B)}$$

M-Step:

$$p(w|\theta_k) = \frac{tf(w,C_{Q+A_k})(1-p_w(z=G))p_w(z=k)}{\sum_{w'} tf(w',C_{Q+A_k})(1-p_{w'}(z=G))p_{w'}(z=k)}$$

where z is a latent variable introduced to represent which type a word is assigned to. $p(z=G)$ and $p(z=k)$ are corresponding probabilities. In this way, we distinguish the *query aspect words* from the *query common words*. The words with high probabilities in θ_k represent the specific query aspect better.

Next, we consider to identify more informative aspect words. We divide the query aspect words into 2 categories: *query dependent aspect words* which provide direct information for the aspect, such as "Tom Hanks" and "Edward Burns" for aspect "actors"; *query independent aspect words* which are query independent and reflect the characteristics of the aspect itself, like "actor", "actress", and "cast" for aspect "actors". We distinguish these 2 types of query aspect words by the assumption that query dependent aspect words occur in C_{Q+A_k}, and query independent aspect words occur in both C_{Q+A_k} and C_{A_k}. The C_{A_k} is the search result of the aspect A_k itself, which contains many words related to the aspect. However, these words can be used for any query with such aspect, but don't bring direct information for a specific query. So we identify query dependent aspect words as $QDW_k = \{t|t \in C_{Q+A_k} \text{ and } t \notin C_{A_k}\}$. The words occur in C_{Q+A_k} that suggests they are related to the query aspect, but don't occur in C_{A_k} that indicates they are query dependent. The relative importance of the query dependent aspect words could be read out from $p(w|\theta_k)$.

4.2.2 Sentence Selection

To summarize aspect A_k for query Q, we extract sentences from the content of the search result documents in $C_{Q+A_k} \cup C_Q$. The candidate sentences are then ranked based on their specificity, redundancy and informativeness. The top ranked sentences are used as a summary for the desired aspect.

Candidate sentence filtering based on specificity. Only part of the sentences within the search result are related to the desired aspect. We select a candidate sentence for a desired aspect only if it is closer to the desired aspect than to any other aspects. To measure this, we classify each sentence to one of the aspects:

$$k^* = \underset{i\in\{1,2,\ldots,K,B,G\}}{\text{argmax}} \prod_{w\in s} p(w|\theta_i) \quad (4)$$

where θ_i is an estimated *query aspect words* model or the *query common words* model or the *global background words* model. A sentence within $C_{Q+A_k} \cup C_Q$ is chosen as a candidate only if k^* equals to k. Thus, all the selected candidate sentences are more specific to the desired aspect.

Sentence clustering. The candidate sentences are selected from multi-documents. Redundancy is particular important. On one hand, the same information conveyed by sentences from different documents indicates its importance. On the other hand, it is not good to show duplicate sentences to users. Due to the above reasons, the candidate sentences are grouped into clusters according to lexical features. We adopt a hierarchical clustering approach. Each single sentence is initiated as a cluster. If two clusters are close enough, they are merged. This procedure repeats until the smallest distance between all remaining clusters is larger than a threshold. Edit distance is used to measure the distance between two sentences. We use $U(s)$ to represent the cluster, which the sentence s belong to. The size of this cluster $U(s).size$ indicates the popularity of this cluster or the redundancy of the information this cluster conveys.

Measuring informativeness. Since informative summaries are expected, we measure the informativeness of a sentence based on:

$$info(s|\theta_k) = (1-\beta) \sum_{\substack{w\in s,\\ w\in QDW_k}} p(w|\theta_k) + \beta \sum_{\substack{w\in s,\\ w\notin QDW_k}} p(w|\theta_k) \quad (5)$$

where QDW_k represents the query dependent aspect words for aspect A_k; β is a parameter to tune the impact of the query dependent aspect words.

Sentence ranking. In each cluster, we select one sentence with highest $info(s|\theta_k)$ as the exemplar to represent the cluster. The exemplars selected from all clusters are ranked according to $Weight_k(s)$:

$$Weight_k(s) = log(1+U(s).size) \times info(s|\theta_k) \quad (6)$$

5. EXPERIMENTS

In the experiments, we assume the query aspects are given and focus on evaluating the quality of generated summaries for query aspects. The data sets we used already contain aspects for each query. Our method and baseline methods take both query and aspects as input.

5.1 Data Sets

To the best of our knowledge, few public data set can be used to evaluate the multi-aspect oriented query summarization. We constructed two data sets from well-known data sources, Wikepedia and TREC. We will introduce the data sets and experimental results in following sections.

5.1.1 Wikipedia Data

Each topic page in Wikipedia is composed of a title and a list of sub sections, which describe the topic from different aspects. For example, the title of a page is "Saving Private Ryan", and the page includes subheadings like "Plot", "Cast" and "Production". In our experiments, we treated the title of a page as a query, the meaningful subheadings (top level) as query aspects. We filtered out the meaningless subheadings like "Notes", "References" and "Further Readings" by rules. We also filtered out pages with less than 3 or larger than 10 aspects to avoid noise. We used the textual content under a subheading as the golden reference for the corresponding aspect. In all, we sampled 1000 pages (queries) from an English Wikipedia dump which was collected in January 2011. The statistics of the sampled data is listed in Table 1.

Table 1: The statistic of Wikipedia data set

Topics	1000
Average Length of Topics (words)	2.15
Average Aspects per Topic	5.15
Average Aspect Length (words)	798

We divided the sampled data into develop set and test set. The develop set containing 100 queries was used for parameter tuning. While the test set, which contains 900 queries, was used for comparing performance of different systems. Note that, since our method uses the search results of a search engine which may give Wikipedia pages as returned documents, we removed Wikipedia pages from the search results when doing experiments.

5.1.2 TREC 2009 Web Track Data

The trec data is widely used for search related experiment evaluation. We use the public available query set of TREC 2009 Web track. One goal of TREC 2009 Web Track is evaluating the search result diversity. The data set includes 50 topics and each topic has 3 to 8 manually edited subtopics to be covered. Each subtopic is a description of an information need. Figure 4 shows an example topic provided by TREC 2009 Web track.

We treated each topic as a query and derived query aspects from its subtopic descriptions by simple rules. We first extracted all nouns from a description. Then we excluded those terms which occur in original query, then used the remaining terms as an aspect. For example, for the query "Obama family tree", "mother information" was used as one aspect. In all, we got 50 queries and 4.9 aspects for each query on average.

5.2 Baselines

The proposed algorithm is denoted as `Q-Composite`. We compare it with 2 baselines. Baseline 1 is based on Ling et al [13], denoted as `Ling-2008`. This method first estimates an aspect prior distribution based on term co-occurrence in the corpus, then integrates the priors into a topic model, finally ranks sentences according to the distance between sentence language model and the aspect models. It is proved very effective for mining faceted summaries on relatively clean and formal data sets, like Gene corpus. But it is not oriented to the web search. Like traditional text summarization tasks, they just use a collection of documents related to the centric topic for summarization. We implemented this method and applied it to the multiple aspect based query summarization as a baseline. The aspect model of `Ling-2008` was estimated on the search result of each original query and the sentences for each aspect were extracted from the search results of both the original query and the composite query, which was the same as the input of our method.The second baseline is based on the top sentences in snippets, which are provided by a search engine for each composite query $Q+A_k$, denoted as `Snippet`. The number of the top sentences depends on the total summarization length limit. Though it is simple, it is very strong. These snippets are selected from the top relevant documents of the composite query, so that they are more likely specific to the query aspect. In addition, most snippet generation algorithms are based on single document summarization method, which tend to extract the sentences containing most relevant terms.

```
<topic number="1" type="faceted">
  <query>obama family tree</query>
  <description>Find information on President Barack Obama's family
  history, including genealogy, national origins, places and dates of
  birth, etc.
  </description>
  <subtopic number="1" type="nav">
    Find the TIME magazine photo essay "Barack Obama's Family Tree".
  </subtopic>
  <subtopic number="2" type="inf">
    Where did Barack Obama's parents and grandparents come from?
  </subtopic>
  <subtopic number="3" type="inf">
    Find biographical information on Barack Obama's mother.
  </subtopic>
</topic>
```

Figure 4: An example topic in TREC 2009 web track.

5.3 Parameter Settings

There are several parameters in our method. We tuned the parameters of our method and baselines on the develop set. In our experiments, λ_G was set to 0.95 in order to get more discriminative words. α_B was set to 0.8 to balance query common information and aspect specific information. The threshold used in sentence merging procedure was set to 0.7. The parameter β was set to 0.0, which means to rank sentences based on query dependent aspect words only. For each composite query, we used the top 50 documents from the search result. The words occurring in less than 3 documents were discarded.

5.4 Experiment Design and Evaluation

Due to the different characteristics of the two data sets, we adopt different evaluation strategies and metrics.

5.4.1 Evaluation on Wikipedia Data

For Wikipedia data, we generate the summaries based on real web data. We send both the original query and the composite queries to a commercial search engine and get the search result documents and snippets. For efficiency, we train the model using the snippets and extract sentences from the content of the documents. We use the ROUGE tool for evaluation on Wikipedia Data. ROUGE is a well-known tool for evaluating both single and multi-document summarization [12]. Basically, it is a recall-like metric. A higher ROUGE value means that more useful information is found. ROUGE-1 metric has been proved highly consistent with human judgements, so we take it for evaluation in our experiments. At evaluating time, the golden reference for each aspect is taken from the content of corresponding subheading in a Wikipedia page. Since the extracted sentences for summarization have different length, we let each system generate top sentences and the first 200, 400 and 600 words are used for evaluation.

5.4.2 Evaluation on TREC 2009 Data

For TREC data set, we generate summaries from the corpus provided by TREC rather than the whole Web, namely the ClueWeb09. Our method depends on the search engine's search result, so we need index ClueWeb09 and build a small search engine. We use a simple ranking function to give search result based on BM25 [16], anchor text and stat-

Table 2: Labeling guide and examples. The query is "Saving Private Ryan" and the aspect is "Actors"

Label	Gain Value (Level)	Description	Examples
Informative and specific	5	The sentence focuses on the desired aspect and provides useful information which can help user to know something about the query aspect.	The actors of "Saving Private Ryan" include Tom Hanks, Tom Sizemor, Edward Burns.
Informative but not specific	4	The sentence conveys multi-aspect information about the query. And it does provide useful information for the desired aspect.	Saving Private Ryan is a 1998 American war film, directed by Steven Spielberg and it follows Tom Hanks as Captain John H. Miller.
Specific but not informative	2	The sentence talks about the desired aspect but doesn't provide much detail information.	Saving Private Ryan Cast and Details on TVGuide.com.
Not about this aspect but about the query	1	It provides some information about some aspects of the query but not related to the desired aspect.	Saving Private Ryan is a 1998 American epic war film set during the invasion of Normandy in World War II.
Not about this query	0	The sentence does not talk about the query.	Alphabetized and searchable index of real and fictional events, cast, and places related to films.

ic rank features. It generates snippets by selecting the top sentences which contain the most query terms.

Since the data does not provide golden reference at sentence level, we have to judge the quality of generated sentences manually. So it is necessary to clarify the standard for assessment. Ideally, a good query summary should make users get the desired information directly. In our scenario, we assess the summaries from two perspectives: specific and informative. First, we hope the summary can give specific information about an aspect rather than a general description covering multiple aspects. Second, it should give more direct information in contrast to navigational information so that users spend less time to obtain information.

Based on this standard, we asked labelers to label the generated sentences for 50 queries. For each system and each query aspect, the labelers had to evaluate the top 3 ranked sentences. Each sentence was assigned a gain value according to the guidelines shown in Table 2, which describes the labeling standard by using an example. Note that we skip the gain value 3, because we think that the "informative and specific" and "informative but not specific" sentences are useful to users for getting direct information, should be given higher bonus than other levels. The topic descriptions, as shown in Figure 4, were also presented to labelers as reference.

The normalized Discounted Cumulative Gain (nDCG) [8] is used to evaluate the performance. The nDCG is a metric that gives higher weights to well ranked objects. The average nDCG over all the test query aspects is used to measure the overall performance.

5.5 Experimental Results and Discussion

In this session, we present the experimental results on two data sets and analyze the performance of different systems and the impacts of key factors.

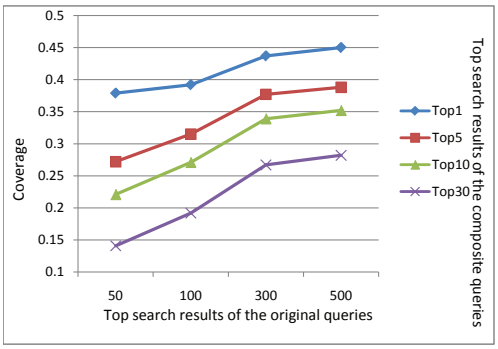

Figure 5: The average coverage of the search results of the original queries over the composite aspect queries.

5.5.1 *Coverage of the Search Results of the Original Queries*

Previous work focuses on organizing the search result of the original query into multiple aspects. We argue that the search result of an original query may not have enough information covering all query aspects. To verify this, we conduct a simple experiment to measure the coverage of the search results of the original queries on the corresponding composite queries. We sampled 100 queries from the Wikipedia data set. For each original query Q, we retrieved the set of top N URLs from a search engine, denoted as S_Q^N. For each composite query $Q + A_k$, we retrieved the set of top M URLs from the same search engine, denoted as $S_{Q+A_k}^M$. We measured the coverage of S_Q^N over $S_{Q+A_k}^M$, i.e., $\frac{|S_Q^N \cap S_{Q+A_k}^M|}{M}$.

The average coverage over all queries' aspects is shown in Figure 5. Intuitively, the search result of $Q + A_k$ should de-

scribe the query aspect better. However, the top documents in $S^M_{Q+A_k}$ rarely appear in S^N_Q. For example, more than 60% top 1 documents retrieved by composite queries are not in the top 100 returned documents for the corresponding original queries. When considering more top documents in $S^M_{Q+A_k}$, the coverage is even smaller. These observations indicate that, at the document level, the search results of the original queries couldn't cover most relevant information related to query aspects. By using composite queries, we could get much more relevant information. Next, we evaluate the quality of the fine-grained information units generated by systems.

5.5.2 System Comparisons

Figure 6 shows the performance comparisons of different systems on Wikipedia test set, varying the word number of summary length limit. We can see that `Q-Composite` outperforms both `Ling-2008` and `Snippet`. The results on TREC 2009 data have the similar trend, which are shown in Figure 7. We have found favorable results for Q-Composite on both NDCG@1 and NDCG@3. This shows proposed method is effective to extract more informative and aspect specific sentences. Especially, `Q-Composite` gains great improvement on NDCG@1, which is important for presenting condensing information on result pages.

To gain more insights, we analyze the label level distributions of the generated summary sentences of the 3 systems on TREC 20009 data set, as shown in Figure 8. The X-axis are label levels. The Y-axis is the distribution. We can see that our method provides more informative sentences (level 5 and level 4) compared with baselines. However, all systems still generate less specific and informative sentences than navigational sentences. This indicates the task is really challenging.

`Q-Composite` performs better than `Ling-2008`. The reasons may include: (1) `Ling-2008` estimates the aspect model on the search result of the original query. There may be not enough information covering all query aspects as shown in section 5.5.1. Therefore, for difficult aspects, it is unable to estimate accurate models. (2) In the search result of the original query, information related to multiple aspects often mixes together. It increases the difficulty to estimate discriminative aspect models. Therefore, it is more difficult to provide specific information for desired aspect. (3) The search result is so noisy that there are many navigational sentences. For example, sentences containing "actors" may also contain words like "cast", "list" and "actress". These words are very easy to have higher weights in aspect models and the sentences are ranked high as well. However, such sentences may only contain navigational information but can't provide direct information. Another reason affecting the performance of `Ling-2008` may be that we did not implement the variation with regularization, which is more complex but reported having better performance than the basic algorithm with Dirichlet model priors. In contrast, by using composite query based method, we are able to get more aspect specific information and roughly classify the information. By distinguishing query dependent aspect words and query independent aspect words, we give bonus to sentences that are aspect specific but also contain more information beyond aspect words.

`Snippet` performs well on Wikipedia data set. It is reasonable, since the snippet generation algorithm favorites the

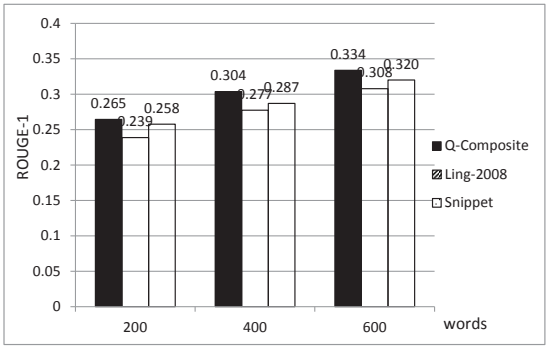

Figure 6: ROUGE-1 performance of Q-Composite and baseline systems on Wikipedia test data.

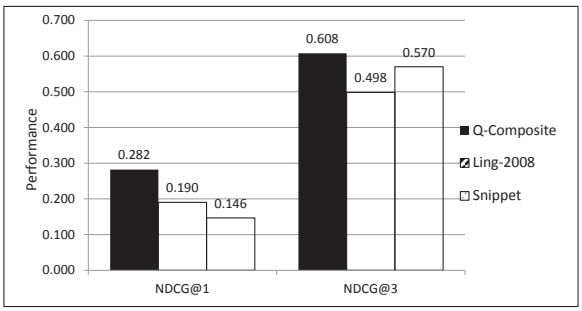

Figure 7: Performance comparisons between systems on TREC 2009 data set.

sentences containing many query terms. Thus the generated summaries match many query aspect terms, which benefits ROUGE-1 metric, especially when the length of summaries is short. However, the snippets don't show much informative information. From the Figure 8, we can see that `Snippet` provides more level 2 sentences (specific but not informative), but very few level 4 and level 5 sentences. The generated sentences usually lack of detail description about the query aspect, mostly are just navigational sentences which often fail to satisfy user information need directly. Our method could get more aspect specific information by comparing the search results of multiple composite queries. Highlighting query dependent aspect words also helps select more informative sentences. `Snippet` generates less irrelevant sentences. One reason is that most sentences in snippets contain original query terms, while other methods don't have such constraint. Another reason may be that using composite queries may lead to topic drift, if the search results of the composite queries contain much noise.

5.5.3 The Impact of Query Dependent Aspect Words

Our method distinguishes the query dependent aspect words and query independent aspect words. We examine the impact of these two types of words. We set the parameter β to be 0.5 in Equation 5 , which means we do not distinguish query dependent and independent aspect words. We denote it as `Q-Composite-AVG`. Since the human judgements on TREC 2009 data directly measure the informativeness of the generated summaries, we compare `Q-Composite` ($\beta = 0.0$) and `Q-Composite-AVG` on this data set. Figure 9 shows the level distributions of the generated top 1 sentences. We can

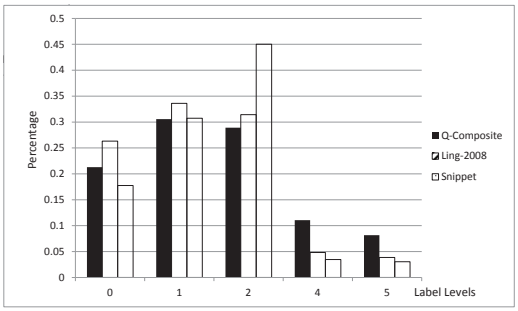

Figure 8: Level distributions of systems on TREC 2009 data set.

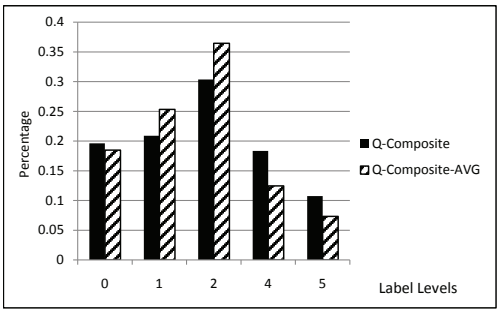

Figure 9: Level distributions of Q-Composite and Q-Composite-Avg on TREC 2009 data set.

see that `Q-Composite` generates more informative sentences (level 4 and level 5). In contrast, the `Q-Composite-AVG` generates more specific but non-informative sentences (level 2). That is because `Q-Composite` favorites the words not only related to the aspect but also related to the original query. The results show that distinguishing query dependent aspect words and independent words is useful for identifying more informative sentences. However, we also see that `Q-Composite` selects slightly more irrelevant sentences. This is because some composite queries bring in more noise, which leads to topic drift.

5.5.4 *The Impact of Search Engine Result*

Our method uses the search results returned by the search engine. In this section, we examine whether the quality of returned documents can affect system performance. We simulate some not very good results, by removing some documents from the search results or randomly picking documents. We test on the Wikipedia test set, since the evaluation can be done automatically. In details, we evenly `remove` 5 documents from the top 50 search results, denoted as `remove5`, namely the 1st, 11st, 21st, 31st and 41st documents. We construct the `remove15` in the same way. We also randomly sample 50 documents from the top 1000 results (denoted as `random`) and select the last 50 documents (denoted as `tail`).

The experimental results are shown in Figure 10. When the search results are not so bad (`remove5` or `remove15`), where most of the documents are relevant, the results are comparable. However, as the relevant documents reduce and noisy data increases, the models may be not very accurate. It shows worse results on `random` and `tail`. The results indicate our method depends on the quality of the search engine search results. For difficult composite queries, there may be no enough relevant candidate sentences for summarization. More noise may lead to topic drift as well.

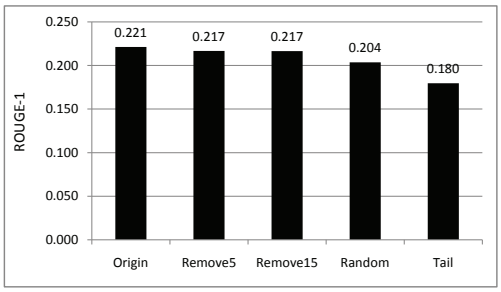

Figure 10: The impact of seach engine, on Wikipedia test set using ROUGE-1 performance.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a multi-aspect oriented query summarization task. This task aims to summarize a query from multiple aspects which are aligned to user intents. Ideally, the users could get relevant information satisfying their information needs directly. Specifically, we formulated the task into 2 main phases: information gathering and summary generation. In the information gathering phase, we proposed a composite query based strategy, which proactively gets information based on the search engine. This strategy differs from traditional search result organization and text summarization, where the set of documents to be deal with is seen as a given system input. In the summary generation phase, we took into consideration the specificity, informativeness and redundancy for sentence selection. We conducted experiments on 2 data sets. Both automatic evaluation and manually judgements were explored. We emphasized that the quality of aspect oriented summaries should be evaluated according to their specificity and informativeness. The experimental results showed that by using composite queries, much more aspect relevant information could be got and our method outperformed 2 baselines for generating informative summaries.

The proposed method attempts to directly provide well organized and relevant *information* to users, as opposed to relevant *documents*. We have several possible directions of future work. First, in this paper we assume the query aspects are given. We would examine the system performance when using automatically mined query aspects. Second, more advanced methods could be exploited to integrate multiple sources of information related to a query for generating more informative summaries. Third, the composite query strategy could be applied for search result diversification by retrieving more aspect related documents.

Acknowledgments

The 1st, 4th and 5th authors are supported by the National Natural Science Foundation of China under Grant No. 60736044, by the National High Technology Research and Development Program of China No. 2011ZX01042-001-001, by Key Laboratory Opening Funding of MOE-Microsoft Key

Laboratory of Natural Language Processing and Speech, Harbin Institute of Technology, HIT.KLOF.2009020.

7. REFERENCES

[1] H. Chen and S. T. Dumais. Bringing order to the web: automatically categorizing search results. In *CHI*, pages 145–152, 2000.

[2] V. Dang, X. Xue, and W. B. Croft. Inferring query aspects from reformulations using clustering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 2117–2120, New York, NY, USA, 2011. ACM.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

[4] Z. Dou, S. Hu, K. Chen, R. Song, and J.-R. Wen. Multi-dimensional search result diversification. In *Proceedings of the 4th ACM WSDM*, pages 475–484, New York, NY, USA, 2011. ACM.

[5] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, pages 40–48, Stroudsburg, USA, 2000.

[6] M. A. Hearst. Clustering versus faceted categories for information exploration. *Commun. ACM*, 49:59–61, April 2006.

[7] M. Hu and B. Liu. Mining and summarizing customer reviews. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *Proceedings of the 10th ACM SIGKDD, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177. ACM, 2004.

[8] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR*, pages 41–48, New York, NY, USA, 2000. ACM.

[9] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In *Proceedings of the 13th international conference on WWW*, pages 658–665, New York, NY, USA, 2004. ACM.

[10] B. Y.-L. Kuo, T. Hentrich, B. M. . Good, and M. D. Wilkinson. Tag clouds for summarizing web search results. In *Proceedings of the 16th ACM WWW*, pages 1203–1204, New York, NY, USA, 2007. ACM.

[11] D. J. Lawrie and W. B. Croft. Generating hierarchical summaries for web searches. In *Proceedings of the 26th ACM SIGIR*, pages 457–458, New York, NY, USA, 2003. ACM.

[12] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the NAACL - Volume 1*, pages 71–78, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[13] X. Ling, Q. Mei, C. Zhai, and B. Schatz. Mining multi-faceted overviews of arbitrary topics in a text collection. In *Proceeding of the 14th ACM SIGKDD*, pages 497–505, New York, NY, USA, 2008. ACM.

[14] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180, New York, NY, USA, 2007. ACM.

[15] A. Nenkova, L. Vanderwende, and K. McKeown. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th Annual International ACM SIGIR, Seattle, Washington, USA*, pages 573–580. ACM, 2006.

[16] S. Robertson and H. Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389, April 2009.

[17] C. Shen, D. Wang, and T. Li. Topic aspect analysis for multi-document summarization. In *Proceedings of the 19th ACM CIKM*, pages 1545–1548, New York, NY, USA, 2010. ACM.

[18] 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 *NTCIR-9 Proceedings*, pages 82–105. Morgan and Claypool, December 2011.

[19] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st ACM SIGIR*, pages 2–10, New York, NY, USA, 1998. ACM.

[20] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Learning query-biased web page summarization. In *Proceedings of the 6th ACM CIKM*, pages 555–562, New York, NY, USA, 2007. ACM.

[21] D. Wang, S. Zhu, T. Li, and Y. Gong. Multi-document summarization using sentence-based topic models. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 297–300, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[22] X. Wang, D. Chakrabarti, and K. Punera. Mining broad latent query aspects from search sessions. In *Proceedings of the 15th ACM SIGKDD*, pages 867–876, New York, NY, USA, 2009. ACM.

[23] X. Wang and C. Zhai. Learn from web search logs to organize search results. In *Proceedings of the 30th annual international ACM SIGIR*, pages 87–94, New York, NY, USA, 2007. ACM.

[24] R. White and R. Roth. Exploratory search. beyond the query-response paradigm. In *Synthesis Lectures on Information Concepts, Retrieval, and Services Series, Gary Marchionini (ed.), vol. 3*. Morgan and Claypool, 2009.

[25] F. Wu, J. Madhavan, and A. Halevy. Identifying aspects for web-search queries. In *Journal of Artificial Intelligence Research*, pages 677–700, 2011 (40).

[26] W.-t. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. Multi-document summarization by maximizing informative content-words. In *Proceedings of the 20th IJCAI*, pages 1776–1782, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[27] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma. Learning to cluster web search results. In *Proceedings of the 27th annual international ACM SIGIR*, pages 210–217, New York, NY, USA, 2004. ACM.