Summarizing Highly Structured Documents for Effective Search Interaction

Lanbo Zhang, Yi Zhang, Yunfei Chen
School of Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064 USA
{lanbo, yiz, ychen}@soe.ucsc.edu

ABSTRACT

As highly structured documents with rich metadata (such as products, movies, etc.) become increasingly prevalent, searching those documents has become an important IR problem. Unfortunately existing work on document summarization, especially in the context of search, has been mainly focused on unstructured documents, and little attention has been paid to highly structured documents. Due to the different characteristics of structured and unstructured documents, the ideal approaches for document summarization might be different. In this paper, we study the problem of summarizing highly structured documents in a search context. We propose a new summarization approach based on query-specific facet selection. Our approach aims to discover the important facets hidden behind a query using a machine learning approach, and summarizes retrieved documents based on those important facets. In addition, we propose to evaluate summarization approaches based on a utility function that measures how well the summaries assist users in interacting with the search results. Furthermore, we develop a game on Mechanical Turk to evaluate different summarization approaches. The experimental results show that the new summarization approach significantly outperforms two existing ones.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

Summarization, Structured Documents, Facets, Facet Selection, Summary Evaluation

1. INTRODUCTION

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

Color:	Color
Country:	USA
Director:	Cameron, James (I)
Distributor:	Twentieth Century Fox Film Corporation [us] - (2014) (USA) (theatrical)
Genres:	Action Adventure Sci-Fi
Language:	English
Related movie:	Avatar (2009)
Release date:	USA 2014
Title:	Avatar 2 (2014)
Url:	http://www.imdb.com/Title?Avatar 2 (2014)

Figure 1: A faceted document example (a movie)

To deal with the information overload problem, search engines help users filter out the majority of useless information by returning documents that are likely to be relevant. However, users still need to judge which documents in the returned results are most useful to them based on the summaries of the retrieved documents. Based on their judgments, users will determine which particular search results they should navigate to. In this perspective, summaries of retrieved documents are important since they will directly influence the user's decision on how to interact with the search results.

Highly structured documents with rich metadata are becoming increasingly prevalent on the Internet and in various verticals. An important trait of highly structured documents is that the major part of a document is composed of metadata. Examples of this type of documents are those representing different kinds of entities, such as products, movies, persons, corporations, etc. In these documents, each metadata field characterizes a specific facet of the entity, and may be assigned with one or several values which are usually short, but contain very important information. In this paper, we use the phrase **faceted documents** to refer to this type of documents, and we call each metadata field a **facet**, a metadata field assigned with a particular value a **facet-value pair**. For simplicity, we sometimes use the term "FVP" to denote "facet-value pair". Figure 1 shows a faceted document example (a movie), where the bold words are different facets, each of which is followed by the value(s) of the facet. In this document, the facet "genre" has three values: "Action", "Adventure", and "Sci-Fi", which correspond to three facet-value pairs respectively: "genre: Action", "genre: Adventure", and "genre: Sci-Fi".

Although search engines for faceted documents are be-

coming increasingly prevalent (e.g., Amazon/eBay/IMDB Search), there has been little research on summarization of faceted documents. There has been a large volume of work on document summarization in the last several decades. However, most of the existing work has been focused on the summarization of unstructured documents. In this paper, we study the problem of summarizing faceted documents in the retrieval context, where the key question is: given a query and a retrieved document composed of a group of facet-value pairs (e.g., Figure 1), how should the system select a small number of facet-value pairs that fit the space (i.e. size) constraint, while delivering as much useful information about the document to the search engine user?

The simplest approach is to manually select a few important facets, and only facet-value pairs of those facets will be included in the summary. This method is query-independent and widely used by many commercial search engines including Amazon Search. For almost all searches on Amazon, the returned product summaries always contain the same facets including title, price, rating, and the shipping information. This solution may not be sufficient, since different users may care about different aspects of a product. Ideally, summaries should be tailored to individual searches. For example, for the query "15-inch silver laptop by Lenovo", the product facets "screen size", "color", "category", and "maker" are very important information to the user, and thus should be included in each product summary so that the user can judge the quality of each retrieved result more accurately.

An alternative approach is to adapt summarization approaches initially developed for unstructured documents (e.g., query-biased approaches) to faceted documents. Most of the existing query-biased summarization approaches select relevant sentences from the original document, and generate the summary by compressing the relevant sentences. To use the existing approaches, we can treat each facet-value pair as a sentence, and use existing sentence-selection approaches to select the best facet-value pairs. However, this approach suffers from two major problems. First, structural information of faceted documents might be ignored. For example, this approach cannot distinguish between different facets, which may not be equally important to the user. Second, since most query-biased approaches tend to select segments/sentences with query terms, some important facet-value pairs without query terms might never be shown in the summary. For example, for the query "15-inch silver laptop by Lenovo", the facet-value pair "color: black" is unlikely to be shown in the summary using existing approaches. However, it is in fact very useful information for the searcher to identify that this document is actually non-relevant.

In this paper, we propose a new summarization approach specifically designed for faceted documents. We observe that a query searching for faceted documents usually implicitly or explicitly involves one or several facet-value pairs that jointly define the information need behind the query. If we can identify the relevant facet-value pairs, we will be able to know which facets might be important for the query, and thus be able to generate better summaries for the user based on those facets. For example, for the query "15-inch silver laptop by Lenovo", we can learn that "color: silver" might be a relevant facet-value pair, thus "color" might be an important facet, and thus we probably should show this facet for all retrieved documents whatever the corresponding value is. In order to discover the relevant facet-value pairs, we propose a learning-based approach for ranking facet-value pairs. To evaluate our approach, we propose a utility-based summary evaluation framework, and compare our summarization approach with two existing ones. The major contributions of this paper include:

1. We propose a new summarization approach for faceted documents based on query-specific facet selection. In particular, we propose a learning-based approach and a set of useful features for ranking facet-value pairs.
2. We argue that a good summary should assist the user in interacting with the search engine. Accordingly, we propose a utility-based evaluation framework for document summarization in the retrieval context.
3. Motivated by research in Experimental Economics, we design a game for crowdsourcing the evaluation of summarization approaches with Mechanical Turk.
4. We compare our new summarization approach with two existing approaches using the game we design on Mechanical Turk.

2. RELATED WORK

Query-biased summarization approaches have been shown to perform better than generic summarization approaches in retrieval tasks. In [23], Tombros et al. compared the query-biased summaries with the static summaries composed of the title and first few sentences of retrieved documents, and found that query-biased summaries can help users improve the speed and accuracy in identifying relevant documents. Similar results were found in [27]. Major search engines including Google, Yahoo, and Bing usually summarize a search result by including the web page title, URL, and a query-biased snippet in the summary [5].

Previous work on document summarization has been largely focused on unstructured documents, where the key question is how to select good sentences from the original document. There has been a large volume of work focused on sentence selection for document summarization [23, 27, 10, 25, 19, 15, 26, 17, 3]. The commonly used attributes of sentences include their positions in the document, the words and query terms they contain, linguistic cues, relationships between sentences, etc. Some approaches take into account the diversity and coverage of a summary while selecting sentences [4, 15].

There has been little work on summarizing structured documents until recently. Huang et al. explored the snippet generation problem in XML search in 2008 [12]. Their approaches are designed based on the assumption that a query result snippet should: 1) be a self-contained and meaningful information unit, 2) be able to differentiate itself from other query results, and 3) be representative of the query result. The snippet retrieval track of INEX 2011 focuses on how best to generate informative snippets for XML search results, in which the Wikipedia corpus is used.

Different from existing work, in this paper, we are focusing on summarizing highly structured documents (i.e. faceted documents) which contain very few texts, and are mainly composed of metadata. Summarization of this type of documents is no longer a sentence selection problem, but a metadata selection problem. To the best of our knowledge, there has been very little work on metadata selection for document summarization.

Traditionally, the evaluation of summarization systems involves measuring quantitative attributes of the summaries, such as the similarity between automatically generated summaries and human-created ones [6, 14, 22]. In 1990s, there had been attempts to develop schemes that measure qualitative features of the systems in a task-based environment [11, 20, 18, 24]. During the last decade, a commonly used scheme for the evaluation of summarization systems has been to ask subjects to do relevance judgments based on document summaries, where the precision, recall, and speed of user judgments, and the number of references to the full document are used as the major metrics [23, 27]. In this paper, we propose a new unified evaluation measure based on the utility of summaries to the user.

There has been some work on selecting relevant facet-value pairs of queries. In [28], Zhang et al. proposed a few heuristic approaches for selecting facet-value pairs from semi-structured documents in a faceted feedback mechanism. In the data-centric track of INEX 2011 [8], one task is to select facet-value pairs of movies for users to provide feedback. Our work is different in that we try to tackle the FVP-ranking problem using a learning-based approach and we propose and study a number of features of different sorts for FVP ranking.

3. EXISTING APPROACHES

In this section, we review two existing approaches that can be used for summarizing faceted documents. The first approach is currently used by commercial search engines, and the other one was initially proposed for summarizing unstructured documents, which can be adapted to faceted documents.

3.1 Manual Facet Selection

In this approach, a fixed set of presumably important facets are manually selected for summarization of all documents. To summarize a document, the facet-value pairs of the selected facets in the document will be chosen to form the summary. There might be different ways to select the fixed set of facets, for example, based on domain knowledge or other considerations such as to maximize click-throughs, purchases, or conversion rates. This approach has been widely used in commercial search engines (such as Amazon Search, IMDB search, etc.).

3.2 Maximum Marginal Relevance (MMR)

A typical approach of query-biased summarization is an incremental sentence selection approach based on the criterion of Maximum Marginal Relevance [4]. At each step, MMR selects a sentence that is similar to the query but dissimilar to the already selected sentences in the summary.

MMR can be adapted to faceted documents by treating each facet or each facet-value pair as an information unit. Algorithm 1 shows the summarization process for a document, and Equation 1 shows how to select the next information unit at each step.

$$\mathrm{u}_{k+1} = \arg \max_{\mathrm{u}_i \in \mathbf{U} \setminus \mathbf{U}_k} \left\{ \lambda * \mathrm{sim}_1(\mathrm{u}_i, \mathrm{Q}) - (1-\lambda) * \max_{\mathrm{u}_j \in \mathbf{U}_k} \{\mathrm{sim}_2(\mathrm{u}_i, \mathrm{u}_j)\} \right\} \quad (1)$$

In Equation 1, u_{k+1} denotes the next information unit we will select, $\mathbf{U}$ denotes the set of all information units in the document, $\mathbf{U}_k$ denotes the set of information units that have been selected in the previous k steps, Q is the user query, sim_1 and sim_2 can be any similarity functions such as cosine similarity, TFIDF, etc., λ is the coefficient to trade off between relevance and diversity.

Algorithm 1 : Summarization based on MMR

Input:
Q: the user query
$\mathbf{U}$: the set of all information units in document d
M: the maximum number of information units allowed
1) Initialize: $k = 0$; $\mathbf{U}_0 = \emptyset$
2) While the size of $\mathbf{U}_k$ is smaller than M
3) Select the next unit u_{k+1} according to Equation 1
4) $\mathbf{U}_{k+1} = \mathbf{U}_k \cup \{\mathrm{u}_{k+1}\}$
5) $k = k + 1$
6) end
Output: the set of selected information units ($\mathbf{U}_k$)

When applied to faceted documents, MMR suffers from two major drawbacks. One is that MMR ignores structural information of documents, which might be crucial for determining the relevance of a document. For example, for the query "movie with Tom Cruise", MMR cannot distinguish between "actor: Tom Cruise" and "producer: Tom Cruise". In fact, the user is more likely to search for movies with Tom Cruise as an actor. The other drawback of MMR (and other query-biased approaches as well) is that some important information units without any query terms are less likely to be included in the summary. In the previous query example, if a movie has no Tom Cruise as an actor, the facet "actor" usually won't be included in its summary although this facet is an important indicator to show that this movie is actually non-relevant.

4. SUMMARIZATION BASED ON QUERY-SPECIFIC FACET SELECTION

In a search application for faceted documents, the information need behind a query is usually related to a group of facet-value pairs. Sometimes, the information need can even be totally represented by one or several facet-value pairs. For example, the query "15-inch silver laptop by Lenovo" can be represented using four facet-value pairs: "category: laptop", "screen size: 15 inches", "color: silver", and "maker: Lenovo". Intuitively, the corresponding facets of those related facet-value pairs ("category", "screen size", "color", "maker" in the example) should be shown in a summary since the relevance of a retrieved document largely depends on its value(s) on those facets. For example, a returned product with "screen size: 12-inch" or "color: black" is obviously non-relevant. A good summary should include those important facets in order to help the user determine the relevance of a document quickly.

Most of the existing query-biased summarization approaches contain two major steps. First, select sentences that are most relevant to the query. Second, build the summary by compressing the sentences to maximize certain criteria (query term coverage, novelty, readability, etc.) while meeting the space constraint [19]. In our approach, we introduce a new step of facet selection for summarizing faceted documents. As we mentioned, the information need behind

a query might be related to some important facets. When summarizing a document, it is helpful to show the important facets, even if those facets of the document do not "look" relevant to the query (e.g., without any query terms). In the previous query example, the facet-value pair "color: black" of a document is very useful information to show that this document is in fact non-relevant. However, by using existing approaches, this facet-value pair is unlikely to be chosen since it does not contain any query terms. Existing approaches do not have the intelligence to predict that the facet "color" is in fact an important facet for this query. In other words, if we are able to learn the important facets for individual queries, we will be able to generate better summaries, which is exactly the goal of our approach.

Our approach has three major steps. First, we use a learning-based approach to rank all facet-value pairs according to their relevance to the query. Second, given the ranked FVPs, we further rank facets according to their importance. Finally, we generate summaries for each retrieved document based on the most important facets we learn in the previous step. The following subsections describe each step in detail.

4.1 Ranking Facet-Value Pairs

Given a query searching for faceted documents, how can we learn the related facet-value pairs? This task can be viewed as an attempt to understand the hidden information need behind a query. In our previous work, we proposed several heuristic approaches for ranking facet-value pairs in the context of semi-structured documents [28]. In this paper, our approach is different in two aspects. First, we are focusing on faceted documents where metadata dominates a document. The ideal approaches for highly-structured faceted documents might be different from those for semi-structured documents where unstructured text is dominating. Secondly, we use a learning-based instead of heuristic approach for ranking facet-value pairs. Compared with heuristic approaches, learning-based approaches generally perform better for the advantage of being able to combine multiple evidences. In fact, some of the features we propose in this paper are equivalent to the best approaches used in [28], and our experimental results show that the performance can be dramatically improved by using a learning-based approach (Table 3).

In our work, to obtain training data for FVP ranking, we hire a human assessor to judge the relevance of facet-value pairs. To combine multiple features, we use the well-known Gradient Boosted Trees (GBT) [7] as the learning model, which has the advantage of being able to handle deep interactions among features, and has been shown to perform well in other tasks such as learning to rank [16] and sentence selection for document summarization [19].

4.1.1 Features

We use a number of features to measure the relevance between a query and a facet-value pair. These features can be categorized into seven categories based on what type of information they depend on. Table 1 summarizes all the features we use.

1) Query Features

This type of feature only depends on the query. We use two features: the query length (number of words), and the average IDF of all query words. The average IDF is used to measure the uniqueness of a query.

2) Facet Features

This type of feature only depends on the facet. The first (F.Type) is a categorical feature that identifies the facet (the number of unique facets is usually small). The second feature (F.NumValues) is the number of unique values the facet has in the whole corpus. The third feature (F.NumOccrs) is the total number of occurrences of all facet-value pairs of this facet in the whole corpus.

3) Value Features

This type of feature only depends on the value. Two features of this type are used: V.Length is the number of words contained in the value, and V.AvgIDF is the average IDF of all value words.

4) FVP Features

This type of feature depends on the facet-value pair as a whole. P.NumDocs is the number of documents containing this facet-value pair. P.IDF is the Inverse Document Frequency of this FVP.

5) Query-Facet Features

This type of feature measures the similarity between the query and the facet. QF.TFIDF is the TFIDF score between the query and the facet name. This feature might be useful based on the intuition that some users might use the facet name to express the faceted constraint of their information need. For example, the query "movies directed by James Cameron" is related to the facet "director".

6) Query-Value Features

This type of feature measures the similarity between the query and the value. We use four features based on four traditional IR scoring methods. QV.BM25 and QV.TFIDF are the BM25 and TFIDF scores between the query and the value. QV.SIDF is different from QV.TFIDF by ignoring the term frequency. QV.CosSim is the cosine similarity score, where the query and value vectors are calculated using the TFIDF weighting method. Comparing these four features, QV.BM25 and QV.CosSim have a penalty mechanism for long values while the other two do not.

7) Query-FVP Features

This type of feature depends on both the query and the facet-value pair, which are mainly based on the frequency of the FVP occurring in the top retrieved documents. QP.DFN measures how many documents in the top N retrieved ones contain the FVP, where we set N = 10, 100, 1000, and the number of all retrieved documents respectively. QP.DFIDFN is the product of QP.DFN and the IDF of the FVP. This group of features might be useful based on the intuition that a facet-value pair occurring frequently in the top retrieved documents while less frequently in the whole corpus are more likely to be relevant to the query.

4.2 Ranking Facets

A facet is more likely to be important to the query if its facet-value pair(s) are relevant to the query. Based on the predicted relevance scores of all facet-value pairs in previous step, we can further rank facets according to their importance to the query. Specifically, we use the following Equation for facet ranking:

$$s(f_i, Q) = \max_{p_j \in \mathbf{P}(f_i)} \{s(p_j, Q)\} \quad (2)$$

where $\mathbf{P}(f_i)$ is the set of all facet-value pairs of facet f_i, $s(p_j, Q)$ is the relevance score of facet-value pair p_j, which is calculated in the previous step.

Table 1: Features for ranking facet-value pairs

Type	ID	Detail
Query	Q.Length	Number of words in the query
	Q.AvgIDF	Average IDF of query words
Facet	F.Type	The facet type (categorical)
	F.NumValues	Number of unique values of this facet
	F.NumOccrs	Number of occurrences of all values of this facet
Value	V.Length	Number of words in the value
	V.AvgIDF	Average IDF of value words
FVP	P.NumDocs	Number of documents containing this FVP
	P.IDF	IDF of this FVP
Query-Facet	QF.TFIDF	TFIDF score between the query and the facet name
Query-Value	QV.TFIDF	TFIDF score between the query and the value
	QV.BM25	BM25 score between the query and the value
	QV.SIDF	Sum of IDFs of the overlapped words between the query and the value
	QV.CosSim	Cosine similarity between the query and the value
Query-FVP	QP.DF10	FVP frequency in the top 10 retrieved documents
	QP.DFIDF10	QP.DF10 * IDF of the FVP
	QP.DF100	FVP frequency in the top 100 retrieved documents
	QP.DFIDF100	QP.DF100 * IDF of the FVP
	QP.DF1000	FVP frequency in the top 1000 retrieved documents
	QP.DFIDF1000	QP.DF1000 * IDF of the FVP
	QP.DFAll	FVP frequency in all retrieved documents
	QP.DFIDFAll	QP.DFAll * IDF of the FVP

4.3 Summarizing Documents

A faceted document (as shown in Figure 1) can be abstracted as a set of facet-value pairs. To summarize a faceted document is thus to answer the following question: which facets or which facet-value pairs should we choose from the original document? In this paper, we focus on facet selection instead of FVP selection based on two considerations. First, a facet is important to show if only there is at least one relevant (to the query) facet-value pair of this facet in the whole corpus, no matter whether the current document contains the relevant facet-value pair(s) or not. Secondly, a summary interface for faceted documents is usually organized by facets with each facet shown in a single line. It seems more natural to generate summaries by facet selection in order to have a better control on the generated summary. For example, it will be easier to control the maximum number of facets in a summary.

4.3.1 Summarization Based on QSFS

Given the ranked facets, we select the most important facets in a document to generate the summary. To determine which values of a selected facet to show, we can use many existing sentence-selection approaches. The whole process for summarizing all retrieved documents of a query is shown in Algorithm 2. To differentiate our approach from Manual Facet Selection (MFS), we will call our approach Query-Specific Facet Selection (QSFS) in the rest of this paper.

Algorithm 2 : Summarization based on Query-Specific Facet Selection (QSFS)

Input:
Q: the user query
C: the whole corpus
D: the set of retrieved documents to summarize
M: the maximum number of facets allowed in a summary
1) Rank all facet-value pairs occurring in **C** based on Q
2) Rank all facets according to Equation 2
3) For each document d in **D**
4) Initialize $\mathbf{S}_\mathrm{d} = \emptyset$
5) Initialize $\mathbf{F}_\mathrm{d}$: the set of facets available in d
6) While the number of facets in $\mathbf{S}_\mathrm{d}$ is less than M
7) $\mathrm{f} = \arg\max_{\mathrm{f}_i \in \mathbf{F}_\mathrm{d}} \{s(\mathrm{f}_i, \mathrm{Q})\}$
8) $\mathbf{S}_\mathrm{d} = \mathbf{S}_\mathrm{d} \cup \{\mathrm{f}\}$
9) $\mathbf{F}_\mathrm{d} = \mathbf{F}_\mathrm{d} \setminus \{\mathrm{f}\}$
10) End while
11) For each facet in S_d
12) Determine which values to show in the summary
13) End for
14) End for
Output: the summary of each document

4.3.2 Integrating MMR and QSFS

MMR and QSFS are two distinct approaches with very different characteristics. MMR aims to include as many relevant information units in the summary, while keeping low redundancy in the summary. To summarize a document, MMR depends only on the unstructured sentences of the document: it does not use any structural information of the document, nor does it use any information from other documents. QSFS aims to discover the hidden important facets of a query by identifying the relevant facet-value pairs. In contrast to MMR, QSFS takes into account a large number of documents and facet-value pairs occurring in the corpus. From this point of view, MMR is a **local** method focusing on the current document, and QSFS is a **global** method focusing on the query and the whole corpus. Given the different characteristics of MMR and QSFS, we expect that their combination will further improve the summary quality.

The combined approach is similar to QSFS (Algorithm 2) except for two positions. First, we use a new method to choose the next facet in line 7) of Algorithm 2, which is shown in Equation 3. $\mathrm{s}_\mathrm{MMR}(\mathrm{f}_i, \mathrm{Q})$ denotes the score of facet f_i based on MMR (Equation 4), $\mathrm{s}_\mathrm{QSFS}(\mathrm{f}_i, \mathrm{Q})$ denotes the score of facet f_i based on QSFS (Equation 2). c is the coefficient to trade off between MMR and QSFS, which can be tuned in practice. Second, we need to change the order of facet selection (line 6-10) and value selection (line 11-13) in order to calculate the MMR score of each facet.

$$\mathrm{f} = \arg\max_{\mathrm{f}_i \in \mathbf{F}_\mathrm{d}} \{c * \mathrm{s}_\mathrm{MMR}(\mathrm{f}_i, \mathrm{Q}) + (1-c) * \mathrm{s}_\mathrm{QSFS}(\mathrm{f}_i, \mathrm{Q})\} \quad (3)$$

The calculation of $\mathrm{s}_\mathrm{MMR}(\mathrm{f}_i, \mathrm{Q})$ (Equation 4) is similar to Equation 1 except that we are treating facets as the basic in-

formation units instead of sentences. In Equation 4, $\mathrm{v}(\mathrm{f}_i, \mathrm{d})$ denotes all the values of facet f_i in document d, which are treated as a single sentence when calculating the similarities.

$$\begin{aligned} \mathrm{s}_{\mathrm{MMR}}(\mathrm{f}_i, \mathrm{Q}) =& \lambda * \mathrm{sim}_1(\mathrm{v}(\mathrm{f}_i, \mathrm{d}), \mathrm{Q}) \\ & - (1-\lambda) * \max_{\mathrm{f}_j \in \mathbf{S}_\mathrm{d}} \{\mathrm{sim}_2(\mathrm{v}(\mathrm{f}_i, \mathrm{d}), \mathrm{v}(\mathrm{f}_j, \mathrm{d}))\} \end{aligned} \tag{4}$$

5. A UTILITY-BASED EVALUATION FRAMEWORK

Evaluation of summaries in search is a very challenging problem that has not been well studied. Some previous work has been using the precision and recall of subjects' relevance judgments [23, 27] as the metrics. However, it's not clear how to trade off between precision and recall when we need to choose the best from several summarization approaches. More importantly, it does not directly measure the utility of the summary for each search engine user.

To measure the utility of the summaries in a search session, we need to examine how the summaries are used by the user. A user guesses which documents in the returned results might be relevant based on the summaries of retrieved documents. The user will skip (not click) documents considered not relevant and navigate to (click) documents considered relevant. If a clicked document is relevant, the user gains some utility. If a clicked document is not relevant, the user incurs some loss due to the waste of time and cognitive efforts. If a skipped document is relevant, the user also incurs some loss for missing useful information.

Accordingly and motivated by the well-known linear utility measure used in the TREC adaptive filtering task [21], we propose a utility-based framework for the evaluation of summarization approaches. Specifically, for a relevant document clicked by the user (i.e. the user guesses it is relevant), the user utility increases by A; for a non-relevant document clicked by the user, the user utility decreases by B; for a relevant document skipped (i.e. not clicked) by the user, the user utility decreases by C; for a non-relevant document skipped by the user, the user utility increases by D. We summarize the parameters of our utility function in Table 2.

Table 2: The utility function

	Document: +	Document: −
User: click (guess +)	A	$-B$
User: not click (guess -)	$-C$	D

The amounts of utility obtained or lost in different cases (A, B, C, D) depend on the specific application and user. The system or the user can adjust the values of A, B, C, D to fit each specific scenario. For example, in an application needs high recalls, we can increase the penalty for missing a relevant document (C); while in an application needs high precisions, we can increase the penalty for clicking on (i.e. misjudging) a non-relevant document (B).

However, it's worth mentioning that this evaluation framework is based on several assumptions. First, we assume a user will read every document summary in the result list. Second, we assume a user can recognize the relevant document after clicking on and accessing the full document. Third, we assume a user will click on a document if the user guesses the document is relevant (+) based on the summary. This assumption is not always true, especially if a user's information need is already satisfied by the summary (e.g., a user searches for a telephone number and the summary contains the answer). However, this assumption is true in many real world scenarios where the utility of a relevant document is realized through actions (e.g. purchasing, reading, etc.) after clicking on the search result. In general, the utility-based evaluation framework seems a better match of the real retrieval scenario, and it provides a single unified measure for comparing different summaries, in contrast to the measures of precision and recall that are hard to tradeoff in practice [23, 27].

6. EVALUATION METHODOLOGY

6.1 Experimental Goals

We design experiments to answer the following questions:

1. How does the proposed summarization approach based on query-specific facet selection perform? Is it better than existing summarization approaches? We will implement and compare four summarization approaches: 1) Manual Facet Selection (MFS), 2) Maximum Marginal Relevance (MMR), 3) Query-Specific Facet Selection (QSFS), and 4) the combination of MMR and QSFS (MMR-QSFS). MFS and MMR are two existing approaches, while QSFS and MMR-QSFS are two approaches we propose.

2. How does the proposed facet-value-pair ranking approach perform? The performance of the FVP-ranking approach will largely influence the quality of the generated summaries. Specifically, we are interested in: 1) the general performance of the learning-based approach, 2) whether the proposed features are useful, and 3) whether the learning algorithm (GBT) combines multiple features effectively.

6.2 The Data Set

Our data set is from the data-centric track of INEX 2010 [9], which consists of: 1) the IMDB data set including 1,594,513 movies; 2) 26 query topics (keywords, description, and narrative) created by the track participants (in the final version); and 3) relevance judgments of query-document pairs.

We refined this data set to make it more suitable for our experiments. To make it easier for subjects to make relevance judgments and without loss of generality, we focus on one genre of documents by using only those documents representing movies, and remove those non-movie documents such as TV series, etc. This leaves a total of 490,075 movies in our data set. Besides, we observed that the relevance judgments provided by the INEX track participants contain some mistakes. To reduce the influence of those wrong relevance judgments, we hired a graduate student to scrutinize all relevance judgments and correct those obvious mistakes. [1]

To prepare the training and test data for our FVP-ranking approach, we ask the same graduate student to do relevance judgments on query-FVP pairs. To obtain FVP candidates,

[1]Visit http://users.soe.ucsc.edu/∼lanbo for the data

all FVPs are ranked based on the BM25 score between the query and the value (feature QV.BM25 in Table 1), and the top 100 FVPs of each query are selected for relevance judging. As a result, we have a total number of 2600 query-FVP relevance judgments, among which there are 148 relevant ones. [2]

6.3 User Study on Mechanical Turk

We use Amazon Mechanical Turk to evaluate the generated summaries. The subjects are asked to guess (i.e. judge) the relevance of each document based on its summary. One problem is that the Turks are motivated by monetary payoffs rather than any information needs, and thus may not make judgments carefully as we hope. Similar problems have been addressed by experimental economists who often work with paid subjects. The general idea of their solution is to incentivize subjects with real monetary payoffs through a game, in which subjects need to do what researchers hope them do in order to maximize their payoffs [13]. We use the same solution in our experiments, and we pay a subject an amount of bonus depending on his/her judging performance. The amount of bonus is calculated based on the utility function described in Table 2, where we set $A = B = 4$ cents, $C = D = 2$ cents.

For each query, we show the subject the keywords, description, and narrative of the topic, and a list of summaries of up to 15 relevant documents and up to 15 non-relevant documents which are ranked highest by our document retrieval algorithm. For each summary, the subject needs to make a choice among three options: "relevant", "non-relevant", or "not sure" (Figure 2). The subject won't get or lose any cents if he/she chooses "not sure". To ensure a reasonably large sample of users, we hire a total number of around 100 subjects on Mechanical Turk. For each subject working on a specific query, we will randomly choose a summarization approach. In our game, we make sure no subject works on different summarization approaches of the same topic. For each combination of a query and a summarization approach, we have 4 subjects to work on it, and the average utility to the 4 subjects will be used to measure the performance of the summarization approach on this particular query.

In our game, we randomly assign 5∼10 queries to each subject. A subject can choose to continue or stop after he/she completes 5 queries. In our experiments, we find almost all subjects completed all 10 queries, and some subjects particularly expressed their great interests in the game by sending emails to the task organizer. Besides, we find the proportion of documents labeled as "not sure" by Turks is very low (less than 5%) among all judged documents. These facts imply that our evaluation approach is quite effective in terms of attracting participants on crowdsourcing websites.

6.4 Evaluation Metrics

6.4.1 Mean Average Normalized Utility (MANU)

We measure a summarization approach based on the average utility its summaries bring to the users, namely, the average amount of bonus the subjects who have worked on this approach earned. Specifically, we propose the **Mean** (across topics) **Average** (across subjects) **Normalized Utility** (**MANU**) as the major metric.

[2]Visit http://users.soe.ucsc.edu/∼lanbo for the data

Title: **Avatar** (2008)
Stars: Murden, Tony; Arad, Natalie; Mills, Rebecca (III)
Writer: Pickles, **James** (II)
* Based on the above summary, do you think this movie is relevant, not relevant, or not really sure?

Figure 2: A Document Summary Example

$$\text{MANU} = \frac{1}{|\mathbf{Q}|}\sum_{q\in\mathbf{Q}}\frac{1}{|\mathbf{U}_q|}\sum_{u\in\mathbf{U}_q}\text{NU}(q,u) \tag{5}$$

In Equation 5, q denotes a query, $\mathbf{Q}$ denotes the set of all queries, u denotes a user (subject), $\mathbf{U}_q$ denotes the set of subjects that work on query q, and NU(q, u) denotes the Normalized Utility of user u on query q, which is calculated as:

$$\text{NU(q, u)} = \frac{\text{U(q, u)} - \text{MinU(q)}}{\text{MaxU(q) - MinU(q)}} \tag{6}$$

where U(q, u) is the total utility (bonus) subject u gets on query q, MaxU(q) and MinU(q) are the maximum and minimum possible utility of query q, namely, the amount of bonus one gets when he/she correctly (incorrectly) judges all documents of query q.

6.4.2 Existing IR Metrics

For the evaluation of our FVP-ranking approach, we use the Mean Average Precision (**MAP**) as the major metric. In the comparison of different summarization approaches, we will also report the performance of each approach in terms of traditional IR metrics including macro **Precision** and macro **Recall**, which are calculated as follows:

$$\text{Precision} = \frac{1}{|\mathbf{Q}|}\sum_{q\in\mathbf{Q}}\frac{1}{|\mathbf{U}_q|}\sum_{u\in\mathbf{U}_q}\text{Precision}(q,u) \tag{7}$$

$$\text{Recall} = \frac{1}{|\mathbf{Q}|}\sum_{q\in\mathbf{Q}}\frac{1}{|\mathbf{U}_q|}\sum_{u\in\mathbf{U}_q}\text{Recall}(q,u) \tag{8}$$

where Precision(q, u) is defined as the ratio of relevant documents among all documents judged as relevant by u, Recall(q, u) is defined as the ratio of documents judged as relevant by u among all relevant documents.

6.5 Summary Constraints

When generating summaries, we use the following constraints for all summarization approaches: 1) at most 3 facets can be included in a summary; 2) the values of each facet cannot be longer than 100 characters (in order to fit in a single line); and 3) each facet can have at most 4 values (so as not to overwhelm users with too many values). In our user study, we organize each document summary into three lines with each facet shown in a single line. Figure 2 shows a document summary example, where the bold words are those occurred in the query.

6.6 Settings of FVP-Ranking Approaches

We use the package from [1] for learning gradient boosted trees. To determine the best number of trees in GBT, we use the best MAP instead of the smallest error on the validation set. Regarding the parameters of GBT, we use the Bernoulli distribution, a maximum of 3000 trees, an interaction depth

of 5, a minimum number of observations of 10 in each tree node, and a shrinkage parameter of 0.01. These parameters are not tuned since the performance is not very sensitive to them [2]. We use 10-fold cross validation to evaluate our approach, and the average performance on all folds will be reported.

6.7 Settings of Summarization Approaches

Four summarization approaches are implemented and compared in our experiments: Manual Facet Selection (MFS), Maximum Marginal Relevance (MMR), Query-Specific Facet Selection (QSFS), and the Combination of MMR and QSFS (MMR-QSFS). For all approaches, we use the first 13 topics for training and parameter tuning, and the remaining 13 topics for test.

Manual Facet Selection (MFS): To ensure a reasonably good performance of this approach, we select facets based on the collected relevance judgments of facet-value pairs. Those facets with most relevant facet-value pairs to queries in the training set are chosen. The top five facets we get in this way are "title", "actor", "director", "keyword", "genre", which are consistent with our common sense on the important facets of movies.

Maximum Marginal Relevance (MMR): For an individual document, we view each facet with the corresponding values as an information unit when applying MMR. Each facet is treated as a bag of words when calculating the similarity in Equation 1. In our experiments, we use the conventional TFIDF method to measure the similarities. To set the parameter λ used in Equation 1, four different values (0.3, 0.5, 0.7, 1) are tried on the training set, and the value that leads to the best summary utility (0.5) is used for testing.

Query-Specific Facet Selection (QSFS): In this approach, we first rank all facets for each query according to Equation 2, then we summarize each retrieved document by selecting the most important facets from the document.

Combination of MMR and QSFS (MMR-QSFS): To set the parameter c in Equation 3, five values (0, 0.3, 0.5, 0.7, 1) are tried on the training set and the best value (0.3) is used for testing. For λ, we use the same value as the one tuned in the MMR approach (0.5).

6.8 More Details

We use the BM25 algorithm implemented in Lemur as the document retrieval approach throughout our experiments. Before scoring a document, we remove all XML tags, and treat it as an unstructured document. We did not use a more sophisticated retrieval method since that is not the focus of this paper.

In our experiments, only the query keywords are allowed to be used for all approaches. The query descriptions and narratives are only used when showing the query to the subjects to help them understand the information need.

7. EXPERIMENTAL RESULTS

In this section, we show the performances of our FVP-ranking approaches, and the performances of different summarization approaches.

7.1 FVP-Ranking Approaches

First, we look at the performances of our FVP-ranking approaches, since the accuracy of the top-ranked FVPs will significantly influence the quality of the generated summaries. Table 3 shows the ranking performance (using four commonly used IR metrics) of each FVP-ranking approach, where GBT is the learning-based approach that integrates all features proposed in Table 1, and all the other 12 approaches are based on individual features[3].

Table 3: Performances of different FVP-ranking approaches. GBT is the learning-based approach, and the other approaches use individual features. GBT significantly outperforms all the other approaches under a paired t-test (p-value < 0.05).

Approach	MAP	R-Prec	P@5	P@R=1
QV.TFIDF	0.18	0.09	0.08	0.11
QV.SIDF	0.18	0.15	0.14	0.10
QP.DF10	0.30	0.25	0.22	0.23
QP.DFIDF10	0.32	0.28	0.22	0.24
QP.DFAll	0.33	0.21	0.30	0.21
QP.DFIDFAll	0.33	0.21	0.30	0.21
QV.CosSim	0.37	0.24	0.33	0.23
QV.BM25	0.42	0.29	0.34	0.28
QP.DF1000	0.48	0.38	0.29	0.36
QP.DFIDF1000	0.51	0.43	0.29	0.38
QP.DFIDF100	0.53	0.43	0.29	0.37
QP.DF100	0.53	0.44	0.29	0.37
GBT	**0.73**	**0.61**	**0.45**	**0.55**

Based on Table 3, we have several findings. First, GBT dramatically outperforms all individual features, which demonstrates the superiority of the learning-based approach, and implies that the features we proposed are quite complementary with each other. Second, QP (Query-FVP) features generally perform better than QV (Query-Value) features, which implies that the frequency among top retrieved documents (measured by QP features) is a stronger signal than the pure text match between the value and the query. Third, among all QP features, QP.DF100 and QP.DFIDF100 perform the best, which implies that 100 might be a reasonable cutoff for top retrieved documents. Fourth, among all QV features, QV.BM25 and QV.CosSim outperform QV.TFIDF and QV.IDF significantly. Note that the major difference between these features is that QV.BM25 and QV.CosSim normalize term frequency based on the value length of the FVP while the other two do not. Given the dramatically different performances, it seems that length normalization is very important for FVP ranking, and this might be generalized to other short-text-ranking problems as well.

7.2 Summarization Approaches

The performances of four summarization approaches are shown in Table 4. Different metrics are reported, while MANU is our major measure. The results show that our proposed approaches (QSFS and MMR-QSFS) are significantly better than the two baselines (MFS and MMR). We discuss each summarization approach in detail in the following subsections.

[3]We didn't use the other features since they do not measure the relevance between a query and an FVP directly, and thus are not suitable for FVP ranking individually.

Table 4: Performances of Different Summarization Approaches. $*$ and $\diamond$ denote a significant improvement over MFS and MMR respectively (p-value $<$ 0.05).

Approach	MANU	Precision	Recall
MFS	0.813	0.871	0.708
MMR	0.887	0.867	0.859$*$
QSFS	0.923$*\diamond$	0.921$\diamond$	**0.887**$*$
MMR-QSFS	**0.929**$*\diamond$	**0.933**$*\diamond$	0.886$*$

7.2.1 Manual Facet Selection (MFS)

Although widely used in commercial search engines, MFS performs significantly worse than the other approaches. This is not surprising since it's not adapted to individual queries. If we take a further look at the Precision and Recall, we find that the poor performance of MFS is mainly due to the low Recall, while the Precision is reasonably good and very close to that of MMR. One possible reason is as follows. Due to the fact that the majority of retrieved documents are non-relevant, subjects tend to be cautious when judging a document as relevant. They usually won't judge a document as relevant unless they observe enough evidences. The major drawback of MFS is that some important facets of a document may not be shown in its summary. As a result, subjects will judge the document as non-relevant by default due to the lack of enough evidences to show that the document might be relevant.

7.2.2 Maximum Marginal Relevance (MMR)

MMR works significantly better than MFS in terms of Recall, which is not surprising given that MMR is more likely to show facets that contain query terms. However, MMR does not perform significantly better than MFS in terms of MANU, and even slightly worse in terms of Precision. There are two possible reasons. First, some important facets without query terms are less likely to be selected by MMR, while they might be selected by MFS. Second, more query terms shown in the summary might mislead the subjects so that they guess the document is relevant even if it is in fact non-relevant, and this might be one possible reason why MMR does not have a good Precision.

7.2.3 Query-Specific Facet Selection (QSFS)

QSFS significantly outperforms both MFS and MMR in terms of the major measure MANU. It's not surprising that QSFS outperforms MFS since QSFS adapts to each individual query and can learn query-specific important facets. However, it's interesting to investigate why QSFS outperforms MMR. One big problem of MMR (and probably most existing query-biased summarization approaches) is that a facet of a document without any query terms is less likely to be shown in the summary, even if this facet is in fact crucial for users to make relevance judgments. QSFS does not suffer from this problem since it aims to discover important facets for each individual query, and the important facets of a document will be shown in the summary no matter whether they contain query terms or not.

Let's consider the query example "movies with Tom Cruise", by which the user is looking for movies acted by Tom Cruise. Assume there is a movie which has Tom Cruise as a producer, and none of the other facets of the movie (including "actor") contain "Tom Cruise". To summarize this movie, MMR will select the facet "producer" but probably not the facet "actor". As a result, there is no evidence in the summary generated by MMR that can verify this document is in fact non-relevant, and some users may even guess this document as "relevant" in order to maximize their utility. Even if some users judge such a summary as "non-relevant" by default, they still could make mistakes. For example, for a movie with Tom Cruise as both an actor and a producer, there is a possibility that MMR will select the facet "producer" but not the facet "actor" in order to ensure a diversity in the summary. QSFS does not suffer from this problem since it will rank both the facet "actor" and "producer" high using our proposed facet-ranking approach.

As we mentioned, the performance of QSFS will largely depend on the performance of the FVP-ranking approach. According to the results reported in Section 7.1, our FVP-ranking approach performs reasonably well on the data set we use, which also explains why QSFS performs well.

7.2.4 Combining MMR and QSFS (MMR-QSFS)

The combination of MMR and QSFS further improves the summarization performance. However, the improvement is not significant. One reason could be that QSFS already works very well on our data set (note that both Precision and Recall are around 90%), so that the space for improvement is limited. Considering the complementary characteristics of MMR and QSFS, the improvement might be more significant on data sets where MMR and QSFS do not perform well individually. We will further explore this approach on more data sets in our future work.

8. CONCLUSIONS AND FUTURE WORK

The task of summarizing highly structured documents in retrieval has not been addressed by prior work, probably due to the lack of a clear definition of good summaries and the lack of an appropriate evaluation methodology. In this research, we assume good document summaries should be informative enough so that a search engine user can judge the utility of the returned results quickly and navigate to the relevant documents without the cost of clicking on many non-relevant URLs/documents. To achieve this goal, a summary should include not only facet-values pairs that match certain query term(s), but also facet-value pairs that make it clear if the underlying document is non-relevant.

We proposed a new summarization approach based on query-specific facet selection, and compared it with a commonly used approach in industry (i.e. MFS) and a well-known approach (i.e. MMR) for summarizing unstructured documents. Also, we proposed a utility-based evaluation framework to measure the effectiveness of summaries in terms of assisting users in interacting with the search results. We developed a game on Mechanical Turk to evaluate the quality of the generated summaries in this framework. The experimental results show that the summaries generated by the new approach are significantly better than those generated by the two baselines. To our knowledge, this is the first paper that focuses on summarizing highly structured documents, which is an important problem given the increasing prevalence of such kind of data.

The work reported in this paper is our first step for sum-

marizing highly structured documents, and can be extended in several directions. For example, the summarization approach proposed in this paper, especially the FVP-ranking component, learns from labeled data instead of user interactions. In the future, we will explore how user interactions can be used to train the model so that the summarization approach can directly optimize the utility measure (i.e. MANU).

It is worth mentioning that two components of our work can be used in other applications besides the summarization of highly structured documents. In order to select important facets, we propose a learning-based approach for ranking facet-value pairs. Specifically, we propose a set of features that are complementary with each other. The experimental results show that the proposed features are useful for FVP ranking, and the learning-based approach can further improve the performance over the best individual feature by integrating multiple features. This approach might be valuable in other applications where FVP-ranking task is involved (e.g., faceted search, faceted query suggestion, etc.). Besides, the utility-based evaluation measure and the crowdsourcing game can be used for the evaluation of unstructured-document summarization as well.

9. ACKNOWLEDGEMENTS

This work was funded by National Science Foundation IIS-0953908, IIS-1144564, UCSC/LANL ISSDM. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsors.

10. REFERENCES

[1] The gbm package. http://cran.r-project.org/web/packages/gbm/.

[2] Generalized boosted models: A guide to the gbm package. http://cran.r-project.org/web/packages/gbm/vignettes/gbm.pdf.

[3] L. L. Bando, F. Scholer, and A. Turpin. Constructing query-biased summaries: a comparison of human and system generated snippets. IIiX '10, 2010.

[4] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. SIGIR '98, 1998.

[5] C. L. A. Clarke, E. Agichtein, S. Dumais, and R. W. White. The influence of caption features on clickthrough patterns in web search. SIGIR '07, 2007.

[6] H. P. Edmundson. New methods in automatic extracting. *J. ACM*, 16:264–285, April 1969.

[7] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):pp. 1189–1232, 2001.

[8] S. Geva, J. Kamps, and R. Schenkel. Inex 2011 workshop pre-proceedings, 2011.

[9] S. Geva, J. Kamps, R. Schenkel, and A. Trotman. Inex 2010 workshop pre-proceedings, 2010.

[10] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: sentence selection and evaluation metrics. SIGIR '99, 1999.

[11] T. F. Hand. A proposal for task-based evaluation of text summarization system., 1997.

[12] Y. Huang, Z. Liu, and Y. Chen. Query biased snippet generation in xml search. SIGMOD '08, 2008.

[13] J. Kagel and A. E. Roth, editors. *The Handbook of Experimental Economics*. Princeton University Press, 1995.

[14] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. SIGIR '95, 1995.

[15] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu. Enhancing diversity, coverage and balance for summarization through structure learning. WWW '09, 2009.

[16] P. Li and C. J. C. Burges. Learning to rank using classification and gradient boosting. NIPS '06, 2006.

[17] D. E. Losada. Statistical query expansion for sentence retrieval and its effects on weak and strong queries. *Inf. Retr.*, 13(5):485–506, Oct. 2010.

[18] I. Mani and E. Bloedorn. Multi-document summarization by graph search and matching. AAAI'97/IAAI'97, 1997.

[19] D. A. Metzler and T. Kanungo. Machine learned sentence selection strategies for query-biased summarization. In *SIGIR 2008 Workshop on Learning to Rank for Information Retrieval*, 2008.

[20] S. Miike, E. Itoh, K. Ono, and K. Sumita. A full-text retrieval system with a dynamic abstract generation function. SIGIR '94, 1994.

[21] S. E. Robertson and D. A. Hull. The trec-9 filtering track final report. 2000.

[22] G. Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summarization. *Inf. Process. Manage.*, 33:193–207, March 1997.

[23] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. SIGIR '98, 1998.

[24] A. Turpin, F. Scholer, K. Jarvelin, M. Wu, and J. S. Culpepper. Including summaries in system evaluation. SIGIR '09, 2009.

[25] R. Varadarajan and V. Hristidis. A system for query-specific document summarization. CIKM '06, 2006.

[26] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Learning query-biased web page summarization. CIKM '07, 2007.

[27] R. W. White, J. M. Jose, and I. Ruthven. A task-oriented study on the influencing effects of query-biased summarisation in web searching. *Inf. Process. Manage.*, 39:707–733, September 2003.

[28] L. Zhang and Y. Zhang. Interactive retrieval based on faceted feedback. SIGIR '10, 2010.