Bridging the Gap between Intrinsic and Perceived Relevance in Snippet Generation

Jing He *PabloDuboue* *Jian-YunNie*
DIRO, University of Montreal
hejing@iro.montreal.ca, dubouep@iro.umontreal.ca, nie@iro.umontreal.ca

ABSTRACT
Snippet generation plays an important role in a search engine. Good snippets provide users a good indication on the main content of a search result related to the query and on whether one can find relevant information in it. Previous studies on snippet generation focused on selecting sentences that are related to the query and to the document. However, resulting snippet may look highly relevant while the document itself is not. A missing factor that has not been considered is the consistency between the perceived relevance by the user in reading the snippet and the intrinsic relevance of the document. This factor is important to avoid generating a seemingly relevant snippet for an irrelevant document and vice versa. In this paper, we incorporate this factor in a snippet generation method that imposes the constraint that the snippet of a more relevant document should also be more relevant to the query. We derive a set of pairwise preferences between sentences from relevance judgments. We then use this set to train a gradient boosting decision tree to model a sentence scoring function used in snippet generation. Compared to the existing snippet generation methods and to the snippets generated by a commercial search engine, our snippets are more consistent with the true relevance of the documents. When the snippets are incorporated into a document ranking function, we also observe a significant improvement in retrieval effectiveness. This study shows the importance to generate snippets indicating the right level of relevance to the search results.

KEYWORDS: Search Engine Snippet, Query-biased Summarization, Document Retrieval.

Proceedings of COLING 2012: Technical Papers, pages 1129–1146,
COLING 2012, Mumbai, December 2012.

1 Introduction

The quality of a search engine is not only determined by its document ranking function, but also by the way the search results presented to the user (Turpin et al., 2009). In particular, good snippets can help the user select relevant documents to click on. Snippets can be considered as a type of extractive summary of the Web pages. It is thus important that a snippet reflects the main part of the document content related to the query topic. This is the traditional role that one assigns to a snippet. To this end, two main categories of approaches have been proposed for snippet generation: sentence retrieval (Sanderson, 1998; Han et al., 2000; Ko et al., 2007; Park and An, 2010; Daumé and Marcu, 2006) and query-biased document summarization (Conroy et al., 2006; Tombros and Sanderson, 1998; Varadarajan and Hristidis, 2005; Otterbacher et al., 2005; Wang et al., 2007; Kanungo et al., 2009).

The sentence retrieval approach finds the most relevant sentences (or text fragments) from a document and uses them as the snippet. One drawback of this method is that it does not consider the fidelity of the sentence to the content of the document, i.e., the extracted sentences may not reflect the main content of the document. There is a high risk of producing false positive snippets leading the user to click on an irrelevant document.

To solve this problem, query-biased document summarization aims to summarize the main content of a document for a specific query. The methods in this category consider both the query and the document. However, an implicit assumption in this method is that the document to be summarized is relevant to the given query. As a matter of fact, the training data used to train the extraction method usually consists of queries and relevant documents. The resulting method could reflect well the relevant part of a relevant document, but may also tend to generate "relevant" snippets for irrelevant documents.

The missing factor in the previous approaches to snippet generation is the consistency between the relevance that the user perceives in reading the snippet (perceived relevance) and the real relevance of the document (intrinsic relevance). The higher-than-intrinsic perceived relevance of snippets leads users to click on irrelevant documents (false positive), while the reverse case leads to not clicking on relevant documents (false negative).

In this paper, we propose a snippet generation method that tries to produce a snippet reflecting the right level of intrinsic relevance of the document. The basic idea is to define a sentence ranking function for a query and a document in such a way that the snippet of a document with a higher intrinsic relevance has a higher perceived relevance. We cast the sentence ranking problem in a learning-to-rank framework (Liu, 2011) with the above constraint. A set of training data is derived from a TREC dataset in which the intrinsic relevance of documents is known. The perceived relevance of a snippet is approximated by a similarity score to the query. A set of features will be defined to reflect different aspects such as the sentence's fidelity to the document and relevance to the query. Finally, we define a cost function for the learning, and propose to use a gradient boosting decision tree model to minimize the cost of the training data set.

When a snippet can reflect the intrinsic relevance of a document, it could be useful to use it to help document ranking. Although some previous studies have examined this by replacing the documents by their snippets for the purpose of increasing efficiency (Sakai and Sparck-Jones, 2001; Wasson, 2002), the results are not conclusive as to whether combining snippets with the

document content can help document ranking. In this paper, we will also incorporate snippets into the document ranking function.

We evaluate the snippet generation methods by comparing the manually judged perceived relevance of the generated snippets to the intrinsic relevance of the corresponding documents, and find that our snippets can better reflect the intrinsic relevance. On F1 score, it improves 14.0% compared to a search engine's snippets, and 7.7% compared to the query-biased summaries. The experiment on document retrieval also shows that our snippets, when incorporated into the document function, are useful in boosting document retrieval.

The remainder of the paper is organized as follows: we first present related work in Section 2. Our snippet generation method is described in Section 3. The experiments are shown in Section 4 and we conclude the paper in Section 5.

2 Related Work

A snippet is usually used as a surrogate for the content of the document in the search engine, and it helps the user determine whether the document contains relevant information (Tombros and Sanderson, 1998; White et al., 2002, 2003).

2.1 Snippet Generation Methods

There are two main categories of methods for generating snippets: sentence retrieval and query-biased summarization.

The sentence retrieval approach finds the most relevant sentences from a document using an IR method, and takes them as a snippet (Sanderson, 1998; Han et al., 2000; Ko et al., 2007; Park and An, 2010; Daumé and Marcu, 2006). One drawback of the approach is that it does not consider the fidelity of sentences to documents, i.e., their relation with the main content of the document. As a consequence, a snippet of an irrelevant document may consist of marginal sentences from the document and appear highly relevant. This problem can be partly tackled by the query-biased summarization method described below.

Query-biased document summarization approaches aim to summarize the content of a document around a specific query. In these approaches, both the document and query are considered in the snippet generation. (Conroy et al., 2006) used query terms and significant terms to select the sentences as snippets. (Tombros and Sanderson, 1998) also considered features such as a sentence's position (title, opening sentence, head, etc.) in weighting. Some work models a document as a graph, in which a sentence is presented as a vertex and the relation between the sentences is presented as a weighted edge. The goal is then to select a minimal spanning tree (Varadarajan and Hristidis, 2005) or a set of central sentences using PageRank-like algorithm as a snippet (Otterbacher et al., 2005).

The query-biased summarization problem can also be thought as a problem of ranking sentences according to their goodness to be a summary, so the learning-to-rank methods such as SVM classifier, SVM-rank (Wang et al., 2007), GBDT (Metzler and Kanungo, 2008) and GBRank (Kanungo et al., 2009) can be used. To build the training data for these methods, human subjects are asked to generate the manual snippet for documents or to judge the snippet's goodness. However, a "good" snippet in this case can reflect the content of a document, but may fail to help users distinguish relevant and irrelevant documents. In this paper, we will incorporate a criterion of consistency between the perceived relevance of the snippets and the intrinsic relevance of the documents in the learning-to-rank framework.

2.2 Snippet Evaluation

It is important to measure the quality of a snippet. The ultimate measure is to put snippet directly in a search task and evaluate how well they can help accomplish the task (Tombros and Sanderson, 1998; White et al., 2003). However, this method is very expensive and not reusable, and the utility measure is influenced by both the retrieval performance and the quality of snippets. Most work targeting the construction of indicative summaries relies on task-based evaluation where the summaries are evaluated as surrogates of the document content (Murray et al., 2009; Kushniruk et al., 2002). Some researchers also proposed to evaluate these two components together (Turpin et al., 2009; He et al., 2010).

Since the snippet generation problem is usually cast as a query-biased summarization problem, evaluation methods in summarization are also used for snippets. In this method, it is assumed that there is a gold-standard summary, and the automatically generated snippet can be compared to it (Wang et al., 2007; Bando et al., 2010). However, (Bando et al., 2010) found this method usually overestimates the quality of the snippets. Alternatively, some work evaluated a snippet by its judged goodness (Kanungo et al., 2009) or readability (Kanungo and Orr, 2009). Users' interaction with the search engine such as clickthrough, dwelling time and eye tracking are also used to evaluate the quality of the snippets (Savenkov et al., 2011; Cutrell and Guan, 2007).

For the task of search, we argue that the quality of a snippet should be primarily evaluated by whether the *perceived relevance* from it can reflect the *intrinsic relevance* of the corresponding document. In SUMMAC evaluation, (Mani et al., 1999) compared the relevance judgments on snippets and on documents, and found the users can make reasonable relevance based on snippets. In the INEX 2011 snippet retrieval task, a measure was used to explicitly examine the consistency between the perceived relevance of the snippets and the intrinsic relevance of the documents (Trappett et al., 2012). However, this measure was only used in the evaluation and none of the participating system used it in the snippet generation process. We believe that we are among the first researchers to use this criterion to train a snippet generator.

2.3 Using Snippets for Retrieval

Since a summary of a document can be considered as a surrogate for the document, it can be used in the document retrieval. Some previous work (Sakai and Sparck-Jones, 2001; Wasson, 2002) tested the utility of snippets for retrieval by replacing the document content by a summary, finding that it can achieve similar precision but worse recall. In this paper, we also test the use of snippets by applying them during the document retrieval. Instead of replacing the document content, we combine the snippet with the document content. This will lead to a higher retrieval effectiveness.

3 Generating Informative Snippets

In this section, we present the method for generating snippets whose relevance is consistent to the documents' relevance. We will first define the problem, and then propose to address this problem in a learning-to-rank framework.

3.1 Problem Definition

(Rose et al., 2007) found that text choppiness and sentence truncation is not good for the readability of search results, and it is better to use complete sentences in snippets. Following

this observation, we define snippet generation as a process of extracting a subset of complete sentences from a document.

A good snippet should of course reflect the main content of the document related to the query topic. It should also provide a good indication on how the document could be relevant, or how two documents compare with respect to relevance. In other words, the comparison between the perceived relevance of two documents should be consistent with their intrinsic relevance. Let's use $R_d(q,d)$ and $R_s(q,S)$ to denote respectively the intrinsic relevance of a document d to a query q and the perceived relevance of the document's snippet S to the query q. The consistency can be defined as follows: Given a query q and a document pair (d_1,d_2) with intrinsic relevance $R_d(q,d_1) > R_d(q,d_2)$, a snippet pair (S_1,S_2) is consistent with the relevance of the documents if and only if $R_s(q,S_1) > R_s(q,S_2)$.

Compared to previous approaches, our additional problem is to find a snippet generation function g^* so that the snippet pairs generated are as consistent with the relevance of the documents as possible. Formally, we need to find a snippet generator g^* that can maximize the expected ratio of the snippet pairs with consistent perceived relevance:

$$g^* = \arg\max_g \{\mathrm{E}_{q,d_1,d_2}[R_s(q,g(q,d_1) > R_s(q,g(q,d_2))|R_d(q,d_1) > R_d(q,d_2)]\} \quad (1)$$

In this paper, a set of judged queries will be used to train g^*. That is, for the training queries, we have document relevance judgments $R_d(q,d)$. The perceived relevance of a snippet $R_s(q,g(q,d))$ will be approximated by a similarity score between the query and the snippet. Even though we used the similarity between queries and snippets to approximate the perceived relevance, the same formalism can use real perceived relevance assessed by human subjects, an approach we might pursue in future work.

We use the learning-to-rank framework to address this problem due to its capability to utilize many different features. The previous work on snippet generation showed that pairwise learning methods perform better than pointwise methods (Kanungo et al., 2009; Wang et al., 2007), and that the methods based one gradient boosting decision tree (GBDT) outperform SVM based methods (Metzler and Kanungo, 2008). In this work, we use GBDT to generate the snippets.

3.2 Gradient Boosting Decision Tree (GBDT)

GBDT is a state-of-the-art learning-to-rank method, which can be learned in a pointwise or pairwise manner (Zheng et al., 2007; Friedman, 2001, 2002; Li et al., 2007; Kanungo et al., 2009; Metzler and Kanungo, 2008).

For pointwise learning, the training data containing N samples can be presented as $\{(x_i,y_i)|x_i \in \mathbb{X}, y_i \in \mathbb{Y}\}_{i=1}^N$, where x_i is a set of extracted features of an item, and y_i is the value of the independent variable, e.g., relevance of a document in the information retrieval task. A ranking function can produce a score for a feature vector $f : \mathbb{X} \to \mathbb{Y}$. Given a loss function $L(y,f(x))$ (we use least-squares loss) defined on the predicted value $f(x)$ and the real value y, the goal is to learn a function f^* in a function class $\mathbb{F}$ that can minimize the sum of loss function on the training dataset, i.e.,

$$f^* = \arg\min_{f\in\mathbb{F}} \sum_{i=1}^N L(y_i,f(x_i)) = \arg\min_{f\in\mathbb{F}} \sum_{i=1}^N \frac{1}{2}(y_i - f(x_i))^2$$

It is an optimization problem, and it can be solved using gradient descent. At iteration k the function $f^{(k)}$ can be updated by $f^{(k+1)}(x) = f^{(k)}(x) - \alpha_k \nabla L(f^{(k)}(x))$, where $\nabla L(f^{(k)}(x))$ is the gradient of $f^{(k)}$ at the point x, and α_k is a coefficient that can be set by line search or at a predefined value.

The gradient descent is implemented in GBDT as follows: At iteration k, we calculate the negative gradient $y_i - f^{(k)}(x_i)$ for each point in the training data, and these points as well as their negative gradients form a new gradient training data $\{x_i, y_i - f^{(k)}(x_i)\}_{i=1}^N$. We can fit a regression decision tree for this gradient training data. For a new point x, we can predict its gradient by the decision tree.

When GBDT is used in pairwise learning (Zheng et al., 2007), the training data is a set of ordered paired items: $\{(x_{i1}, x_{i2}) | y_{i1} > y_{i2}\}_{i=1}^N$. It is expected to produce the predicted values whose pairwise ranking is consistent with that in the training data. We can define a least-squares loss function as follows

$$L(f) = \sum_{i=1}^{N} \frac{1}{2} (\max\{0, f(x_{i2}) - f(x_{i1}) + \sigma\})^2, \sigma > 0 \tag{2}$$

In this formula, the loss is zero when $f(x_{i1}) \geq f(x_{i2}) + \sigma$. A small positive value σ is used to prevent from learning a constant function which produces an identical value for all the points.

Similarly, we can use gradient descent method to solve the problem of optimizing $L(f)$ in this pairwise setting. The negative gradient of one pair of items in the training data can be calculated as:

$$-\nabla L(f^{(k)}(x_{i1}), f^{(k)}(x_{i2})) = \begin{pmatrix} \max\{0, f^{(k)}(x_{i2}) - f^{(k)}(x_{i1}) + \sigma\} \\ \max\{0, f^{(k)}(x_{i1}) - f^{(k)}(x_{i2}) - \sigma\} \end{pmatrix} \tag{3}$$

That is, for one sample of pair of items, we add two samples in the gradient training data $(x_{i1}, \max\{0, (f^{(k)}(x_{i2}) - f^{(k)}(x_{i1}) + \sigma)\})$ and $(x_{i2}, \max\{0, (f^{(k)}(x_{i1}) - f^{(k)}(x_{i2}) - \sigma)\})$, aiming to increase $f^{(k+1)}(x_{i1})$ if $(f^{(k)}(x_{i1}), f^{(k)}(x_{i2}))$ is inconsistent with (y_{i1}, y_{i2}).

3.3 Learning GBDT for Ranking Sentences

In this section, we will propose a method to rank the sentences using the GBDT model.

3.3.1 Training Data

We need training data to learn an informative snippet generator, i.e., $R_d(q, d)$ for a set of queries $\{q\}$ and a set of documents $\{d\}$. Ideally, we would also like to have $R_s(q, S)$ for a snippet S, but this is usually unavailable given the large number of possible snippets for a document. In our implementation, we approximate the perceived relevance by the query-sentence similarity.

3.3.2 Loss Function

Since our objective function is defined on preferences, we can transform the relevance judgments in the training data to a preference data set: $\{(q, d_1, d_2) | q \in Q, d_1 \in D_q, d_2 \in D_q, R_d(q, d_1) > R_d(q, d_2)\}$, where D_q is the set of retrieved documents for q. For a snippet generation function g, we can get a pair of snippets for a given sample of the training data:

$(g(q,d_1),g(q,d_2))$. The perceived relevance is $(R_s(q,g(q,d_1)),R_s(q,g(q,d_2)))$. Similarly to the general GBDT pairwise loss function (Eq 2), we can define the loss function here as

$$L(g)=\sum_{q,d_1,d_2}\frac{1}{2}(\max\{0,R_s(q,g(q,d_2))-R_s(q,g(q,d_1))+\sigma\})^2,\sigma>0 \quad (4)$$

3.3.3 Sentence Scoring Function

It is difficult to train the snippet generator function g directly. Instead, we define a sentence scoring function, and then derive the snippet generator function from the sentence scoring function. A sentence scoring function can be defined as $f:\mathbb{X}\rightarrow\mathbb{R}$, where the input is a vector of extracted features about a sentence s in a document d for a query q, and the output is a predicted score for the sentence. Given the sentence scoring function, the snippet generator function g can be applied to a document d and a query q by: (1) ranking the sentences $s\in d$ by their scores, and (2) selecting the top-n sentences as the snippet up to some predetermined length. Thus, the problem of learning a snippet generation function g can be cast as learning a sentence scoring function f.

3.3.4 Learning Method

We use the GBDT method to learn a sentence scoring function in an iterative manner. At iteration k, we have the sentence scoring function $f^{(k)}$ and the derived snippet generator function $g^{(k)}$. In each iteration, there are two cases:

- Case 1: If the perceived relevance of the snippet of document d_1 is larger than that of d_2, the sentence ranking function should not be modified;
- Case 2: If the perceived relevance of the snippet about document d_2 is larger, then we should modify the sentence ranking function so that the snippet of d_1 becomes more relevant and that of d_2 less relevant;

In Case 2, the modification of sentence ranking function f is achieved by adding new sentence preference pairs into the training set. In particular, for document d_1, we should rank higher the sentences that are more relevant than the snippet of d_2:

$$S^{(k)+}_{q,d_1}=\{s|s\in d_1',R_s(q,s)>R_s(q,g^{(k)}(q,d_2))$$

In our implementation, d' is a subset of d_1 composed by the sentences that contain at least one non-stop query term. As such, we have a preference dataset about d_1:

$$\{(s_1,s_2)|s_1\in S^{(k)+}_{q,d_1},s_2\in g^{(k)}(q,d_1)\}$$

Therefore, the gradient training set for the sentence scoring function can be defined as:

$$(x_1,f(x_2)-f(x_1)+\sigma),(x_2,f(x_1)-f(x_2)-\sigma)$$

where x_1 and x_2 are feature vectors for s_1 and s_2 respectively. Similarly, for the sentences in d_2, we should create

$$\{(s_1,s_2)|s_1\in S^{(k)-}_{q,d_2},s_2\in g^{(k)}(q,d_2)\}$$

where

$$S_{q,d_2}^{(k)-} = \{s | s \in d_2', R_s(q,s) < R_s(q, g^{(k)}(q,d_1))$$

The above process for each iteration has a complexity of $O(|Q| \cdot |D|^2 \cdot |d|)$, where $|Q|$ is the number of queries, $|D|$ is the number of retrieved documents, and $|d|$ is the number of sentences in a document. This complexity is high. We use the more efficient stochastic GBDT method (Friedman, 2002). The algorithm is presented in Algorithm 1.

Input: Parameters: N, M, σ, α
Training Data: Query Set Q, Document Set $\{D_q | q \in Q\}$, Intrinsic Relevance Judgment $\{R_d(q,d) | q \in Q, d \in D_q\}$, Perceived Relevance Judgment $\{R_s(q,s) | q \in Q, s \in d, d \in D_q\}$.
Output: Sentence Scoring Function: f
begin
  Initial $f^{(1)}$ as a constant function
  for $k = 1, \ldots, N$ **do**
    GD = {}
    for $j = 1, \ldots, M$ **do**
      Sample a query $q \in Q$, a pair of documents $d_1, d_2 \in D_q, s.t. R_d(q,d_1) > R_d(q,d_2)$
      Generate snippets $g^{(k)}(q,d_1)$, $g^{(k)}(q,d_2)$
      if $R_s(q, g^{(k)}(q,d_1) > R_s(q, g^{(k)}(q,d_2)$ **then**
        Sample a sentence $s_{11} \in g^{(k)}(q,d_1)$ and a sentence $s_{12} \in d_1 - g^{(k)}(q,d_1)$
        GD.add($\{(x_{11}, 0), (x_{12}, 0)\}$)
        Sample a sentence $s_{21} \in g^{(k)}(q,d_2)$ and a sentence $s_{22} \in d_2 - g^{(k)}(q,d_2)$
        GD.add($\{(x_{21}, 0), (x_{22}, 0)\}$)
      else
        Sample a sentence $s_{11} \in S_{q,d_1}^{(k)+}$ and a sentence $s_{12} \in g^{(k)}(q,d_1)$
        GD.add($\{(x_{11}, f(x_{12}) - f(x_{11}) + \sigma), (x_{12}, f(x_{11}) - f(x_{12}) - \sigma)\}$)
        Sample a sentence $s_{21} \in S_{q,d_2}^{(k)-}$ and a sentence $s_{22} \in g^{(k)}(q,d_2)$
        GD.add($\{(x_{21}, f(x_{22}) - f(x_{21}) + \sigma), (x_{22}, f(x_{21}) - f(x_{22}) - \sigma)\}$)
      end
    end
    Fit a regression decision tree $t^{(k)}$ to the gradient training data GD
    Update the sentence scoring function $f^{(k+1)} = f^{(k)} + \alpha t^{(k)}$ and the corresponding snippet generator function $g^{(k+1)}$
  end
  $f = f^{(N+1)}$
end

Algorithm 1: Training GBDT Sentence Scoring Function

3.3.5 Features

We use four categories of features in the learning methods (see Table 1): Query-Sentence Relevance (QSR), Document-Sentence Fidelity Features (DSF), Sentence Informativeness Features (SI), Query-Document Relevance Features (QDR). The features about the relevance between the query and the document (QDR) are not commonly used in query-biased summarization.

Category	Description	Features
QSR	the relevance between the query and the sentence	- Cosine similarity of TFIDF vectors
DSF	the fidelity of the sentence's content to the document	- Cosine similarity of TFIDF vectors - Number of significant words - Is the sentence the title/heading?
SI	how much information contained in the sentence	- KL-divergence to the collection model - Averaged IDF of the words
QDR	the relevance between the document and the query	- Cosine similarity of TFIDF vectors - BM25 score

Table 1: Features

Since our goal is to generate snippets that can reflect the query-document relevance, these features can help in snippet generation.

4 Experiments

We carried out two experiments to test our snippet generation method. The first experiment compared the perceived relevance of the generated snippets with the intrinsic relevance of the documents. The second experiment use the generated snippets in document retrieval.

4.1 Experiment Setup

4.1.1 Dataset

We use the dataset of TREC Web track 2010 and 2011. There are 100 queries in the dataset (98 of them have relevance judgments), and these queries were extracted from search engine logs. For each query, we were given a short query string, a detailed description and a list of description about the subtopics. We use short queries in the study. The document collection is ClueWeb09B, which contains 428M documents. The dataset also contains relevance judgments. The relevance judgment for a (query, document) pair is in grades {0,1,2,3}, with larger values indicating higher relevance. In our experiment, we took the 50 queries of TREC 2011 for training the sentence ranking function, and the other 48 queries of TREC2010 as test. The information of the training dataset is shown in Table 2. The test dataset will be further pruned due to lack of some baseline results and selection for judgment, discussed in Section 4.1.2 and Section 4.1.3.

Statistics	Dataset		
	Training	Test	Pruned Test
#query	50	48	47
#document	13081	15849	754
#sentence/doc	64.2	66.4	51.1
#word/sentence	10.1	10.3	12.0

Table 2: Experimental Dataset

4.1.2 Tested Snippet Generating Methods

Two baselines are used in our experiment: a commercial search engine (SE) and a query-biased summarizer (SUM). The use of a commercial search engine is to see how other snippet generation methods compare to the current state-of-the-art search engines. In order to collect the snippets for a query-document pair, we send a search query containing the query string and the site of the document's URL. For example, for the query "horse hooves" and the document whose URL is "http://www.snopes.com/military/statue.asp", we construct a query "horse hooves site:www.snopes.com". We then look for the result whose URL is exactly same as the document's URL. The content of some Web pages may change, and search engine generates the snippet based on a document with different content. So we only keep the documents whose content is very similar (cosine value > 0.95) to that in ClueWeb09B dataset.

For many (query, document) pair, we cannot find the snippets generated by the search engine. For a fair comparison, we only use the (query, document) pairs whose snippets can be found in the search engine in our evaluation, and the statistics of this pruned test dataset is shown in Table 2. We also limited the length of generated snippets comparable to the search engine's snippets, the average snippet length of SE, SUM, our method is 153, 130, 142, respectively.

For query-biased snippets, we used the MEAD summarizer (Radev et al., 2004). MEAD clusters sentences and chooses sentences most similar to the centroid of the cluster, discarding sentences too similar to already picked sentences. We also can use an external query file as an extra feature for ranking. For our snippet generation method, we approximate the perceived relevance of the snippet by the query-sentence similarity. Specifically, we use cosine similarity between the TFIDF vectors of sentences and queries. There are four parameters in the learning method (see Algorithm 1): iteration number N, sample size M for each iteration, the predefined cost for identical predicted relevance σ and the shrinking parameter α. We tune these parameters by five-fold cross validation on the training dataset and get $(N = 200, M = 10,000, \sigma = 0.01, \alpha = 0.2)$.

4.1.3 Judgments

For earh query in the dataset, we randomly select up to 20 Web pages and generate the snippets for them (Some topics may have less than 20 documents).

We ask human assessors to judge the perceived relevance of these snippets. For each query, the assessors were given the query string, the description and a list of subtopics. For each snippet, the assessors were asked whether they thought the document behind the snippet can provide relevant information for the query. Each snippet is judged as "relevant" or "irrelevant". Every snippet and query pair was judged first by two assessors, and a third assessor was invited to judge if the previous two assessors disagree. The final judgment used for the evaluation is determined by voting. We found the agreement rate of the first two assessors quite high (86.7%), and the Cohen's kappa value is 0.699.

4.1.4 Measures

We use the evaluation measures used in the INEX 2011 snippet retrieval task (Trappett et al., 2012). Denote the number of true positive, false positive, true negative and false negative snippets for all the test queries are TP, FP, TN, FN respectively. Recall (R) and negative recall (NR) reflect the percent of the relevance/irrelevant documents that can be detected by their snippets. The geometric mean score (GM) of these values are used as the primary measure

in INEX 2011 snippet retrieval task. In addition, the precision (P), F-1 values (F1) and the accuracy (ACC) are used for the evaluation. All measured used are shown in Table 3

Measure	Definition
Precision	$P = \frac{TP}{TP+NP}$
Recall	$R = \frac{TP}{TP+FN}$
F1 Rcore	$F1 = \frac{R \cdot P}{R+P}$
Negative Recall	$NR = \frac{TN}{FP+TN}$
GM	$GM = \sqrt{R \cdot NR}$
Accuracy	$ACC = \frac{TP+TN}{TP+FP+TN+FN}$

Table 3: Definition of Measures

4.2 Results

The results are shown in Table 4. Each row corresponds to one measures, and each column one snippet generation method. For the our snippet generation methods, we also show in the parentheses the improvement rate compared to the baselines (SE and SUM). We found that the scores of our method are higher than both the search engine's snippets and the query-biased document summarization on most measures with an exception on recall. The query-biased summarization method also performs better than the search engine's snippet.

Measure	Snippet Generation Method		
	SE	SUM	Our Method
P	0.412	0.506	0.531(28.9%,4.9%)
R	0.605	0.539	0.596(-1.5%,10.6%)
F1	0.493	0.522	0.562(14.0%,7.7%)
ACC	0.617	0.693	0.714(15.7%,3.0%)
NR	0.622	0.764	0.782(25.7%,2.4%)
GM	0.613	0.642	0.676(10.3%,5.3%)

Table 4: Evaluation Results for the Snippet Generation Methods

For the snippets generated by the search engine, its recall value is the highest of the methods tested, but its precision score is the lowest. In other words, the snippet generator of search engines tends to select sentences with higher perceived relevance, and this way attract the users to click on the snippets. Specifically, 30.8% of the documents within the test collection are relevant, but 44.8% of the snippets from the search engine are assessed as relevant, compared to 33.0% and 34.5% from SUMM and our method, respectively.

For the snippets generated by SUM and our method, the negative recall value is much larger than that of the search engine. This means that those methods can better reflect irrelevant search restuls. Both our method and SUM methods perform worse than SE for the recall metric, but the our method's performance doesn't drop so much (-1.5% and not significant worse) compared to the others.

As an illustration, Table 5 shows two examples of generated snippets. The second column in the table is the labeled intrinsic/perceived relevance. SUM method generates a false-negative

snippet in the first example, and SE and SUM generate false positve snippets in the second example. In both cases, the snippets generated by our method can better indicate if the document is relevant.

Query	-	horse hooves (about horse hooves, care and disease)
Doc	1	clueweb09-en0009-75-38689 (horse hooves cleaning)
SE	1	Cleaning your horses hooves is easy ... Cleaning your horse's hooves regularly will give you a chance to monitor the health of the foot,check the shoes...
SUM	0	Put a halter on your horse and secure him in cross ties or tether him to a rail using a quick release knot. If your horse doesn't pick up his foot...
Our Method	1	How To Clean Your Horses Hooves. This procedure can be used on foals, along with tapping the hoofpick on the bottom of the hoof to prepare the young...
Query	-	discovery channel store (locations, information and produdcts of Discovery Channel stores)
Doc	0	clueweb09-en0011-53-07816 (about a company rebranding for a product by Discovery Channel)
SE	1	Discovery Channel Stores-Ultimate Science Tools 2005 ReBrand 100: merit Order of information below: Before Image, Rebrand Images, Summary...
SUM	1	Challenge-As Discovery Channel changed the face of science and nature programing, Discovery Channel Stores altered the landscape of educational...
Our Method	0	Result-Delivering profit margins that only proprietary products can, the line brands Discovery Stores as the destination for quality, vibrant...

Table 5: Two Examples of Snippets

We have used four categories of features in our experiment in Section 3.3.5. Here we examine the importance of these features. The importance of each category of feature can be evaluated by testing the loss after removing a category of features. If the loss increases much after removing a category of features, it indicates that these features are important for learning. The results are shown in Figure 1. It shows that these four categories of features are all useful for learning our snippet generator. In particular, the features that reflect the relevance between queries and documents (RQD) should be taken into account in generating snippets.

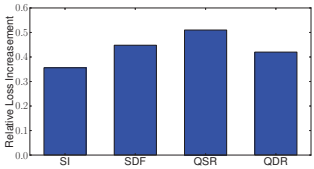

Figure 1: Feature Importance in Our Snippet Generating Method

4.3 Using Snippets for Document Retrieval

In this section, we investigate the role of snippets in document retrieval. We have found that the our snippets can help the users to determine the intrinsic relevance of the documents. Then a natural question is whether the snippets can also be useful to help a search engine determine the relevant documents.

In this experiment, we use a language modeling approach to IR, in which the document model is extended by the snippets as follows:

$$Pr(t|\theta_{d+S}) = \alpha Pr(t|\theta_{d_{\text{LM}}}) + \beta Pr(t|\theta_{S_{\text{LM}}}) + (1-\alpha-\beta)Pr(t|\theta_C)$$

where $\theta_{d_{\text{LM}}}$ and $\theta_{S_{\text{LM}}}$ are maximum likelihood model for the document and the snippet respectively, θ_C is the collection language model for smoothing, α and β are smoothing parameters. These parameters are tuned by grid search in our experiment. Given the new document language model, we use KL-divergence to score the document for a query:

$$KL(\theta_q||\theta_{d+S}) = \sum_t Pr(t|\theta_q) \log \frac{Pr(t|\theta_q)}{Pr(t|\theta_{d+S})}$$

The retrieval performance of the different snippet generation methods is shown in Table 6. The numbers in the parentheses show the improvement compared to retrieval based on the document model without combining with the snippet (NO). Since the snippets of only a subset of documents can be obtained from the search engine, we use the relevance judgments of this subset only in our evaluation. We also test the statistical significance using paired t-test, and † and ‡ indicate the difference is significant with p-value 0.1 and 0.05 respectively.

We can see that the snippets by SUM and our methods can significantly improve the retrieval effectiveness. This result clearly indicates that good snippets can be used to help document ranking. The improvement on top-ranked documents (ERR@10 and NDCG@10) is higher than MAP. This means that these methods have a higher impact on top retrieval results. On the other hand, we did not observe a significant difference when the snippets of the search engine are used. Overall, the rank of the retrieval performance with different snippet generation methods is the same as the perceived relevance accuracy: our method > SUM > SE.

However, we did not observe a significant difference between SUM and our method in IR, as in the experiment on perceived relevance. A possible reason is that both methods generate snippets that are related to the query and to the main content of the document, and the subtle difference is perceived by humans, but not by the simple retrieval model based on bag of words. It is possible that using a more sophisticated retrieval model, the difference between the two methods can be materialized on retrieval effectiveness. This is part of our future work.

Our experiment is different from that of (Sakai and Sparck-Jones, 2001; Wasson, 2002) in which snippets were used as replacement of documents. Their results showed that the retrieval recall decreases while precision remains in doing this. This is predictable as snippets are much shorter than documents, thus contain less less information. However, the short segment of text selected in a snippet is usually highly related to the query. They may provide useful position information about the document content, which can hardly emerge if we look at the whole content of the document. This may be the reason why snippets helped in our experiment on IR.

5 Conclusion and Future Work

In this paper, we addressed the problem of generating informative snippets for search results. A snippet is deemed informative not only because it reflects the part of content of the document related to the query topic, but also because it provides useful indication about the document's relevance. Generating a snippet that is perceived relevant for an irrelevant document is misleading, so is the opposite situation. We specifically addressed the problem of consistency

Measure	Snippet Generation Method			
	NO	SE	SUM	Our Method
MAP	0.328	0.328(0.0%)	0.335(2.1%†)	0.336(2.4%†)
ERR@10	0.109	0.112(2.8%)	0.114(4.6%†)	0.118(8.3%‡)
NDCG@10	0.290	0.293(1.0%)	0.304(4.8%‡)	0.309(6.5%‡)

Table 6: Retrieval Performance of Taking the Snippet as a Field

between the perceived relevance of the snippet and the intrinsic relevance of the document. To our knowledge, this is the first time that such a criterion is incorporated into a snippet generation process.

In this paper, we cast the snippet generation problem as the one to learn a sentence ranking function. The training data are pairwise sentence preference pairs, which are created according to the consistency between their perceived relevance and the real relevance of the documents. We used gradient boosting decision trees to model the sentence ranking function. Our experiments showed two facts: 1) Our method that generates snippets by considering the consistency criterion can provide better indication on the relevance of the documents to users; 2) The snippets of search results can provide useful information for document ranking.

We have explored one possible avenue to generate information snippets using a pairwise learning-to-rank method. Other methods could be explored in the future, such as SVM-rank, etc. Our focus in this paper was on consistency, and we used a simple method to determine the candidate sentences to be included into the training data. It may be better to select sentence candidates by also considering their fidelity to the document and relevance to the query.

On the use of snippets in document ranking, we used a bag-of-word approach, which failed to cope with the subtle differences between snippets. Those differences may emerge in the retrieval results if we use more sophisticated retrieval methods, e.g., by considering the possible relations between terms and by considering more complex units such as phrases. These are the problems that we will address in our future work.

References

Bando, L. L., Scholer, F., and Turpin, A. (2010). Constructing query-biased summaries: a comparison of human and system generated snippets. In *Proceedings of the third symposium on Information interaction in context*, IIiX '10, pages 195–204, New York, NY, USA. ACM.

Conroy, J. M., Schlesinger, J. D., and O'Leary, D. P. (2006). Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 152–159, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cutrell, E. and Guan, Z. (2007). What are you looking for?: an eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 407–416, New York, NY, USA. ACM.

Daumé, III, H. and Marcu, D. (2006). Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 305–312, Stroudsburg, PA, USA. Association for Computational Linguistics.

Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Friedman, J. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.

Han, K.-S., Baek, D.-H., and Rim, H.-C. (2000). Automatic text summarization based on relevance feedback with query splitting (poster session). In *Proceedings of the fifth international workshop on on Information retrieval with Asian languages*, IRAL '00, pages 201–202, New York, NY, USA. ACM.

He, J., Shu, B., Li, X., and Yan, H. (2010). Effective time ratio: A measure for web search engine with document snippet. In *AIRS '10: proceeding of the Sixth Asia Information Retrieval Societies Conference*.

Kanungo, T., Ghamrawi, N., Kim, K. Y., and Wai, L. (2009). Web search result summarization: title selection algorithms and user satisfaction. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1581–1584, New York, NY, USA. ACM.

Kanungo, T. and Orr, D. (2009). Predicting the readability of short web summaries. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 202–211, New York, NY, USA. ACM.

Ko, Y., An, H., and Seo, J. (2007). An effective snippet generation method using the pseudo relevance feedback technique. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 711–712, New York, NY, USA. ACM.

Kushniruk, A. W., Kan, M.-Y., McKeown, K., Klavans, J., Jordan, D., LaFlamme, M., and Patel, V. L. (2002). Usability evaluation of an experimental text summarization system and three search engines: implications for the reengineering of health care interfaces. In *Proc AMIA Symp. 2002*, pages 420–424.

Li, P., Burges, C. J. C., and Wu, Q. (2007). Mcrank: Learning to rank using multiple classification and gradient boosting. In *NIPS*.

Liu, T.-Y. (2011). *Learning to Rank for Information Retrieval*. Springer.

Mani, I., House, D., Klein, G., Hirschman, L., Firmin, T., and Sundheim, B. (1999). The tipster summac text summarization evaluation. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, EACL '99, pages 77–85, Stroudsburg, PA, USA. Association for Computational Linguistics.

Metzler, D. and Kanungo, T. (2008). Machine learned sentence selection strategies for query-biased summarization. In *SIGIR Learning to Rank Workshop*.

Murray, G., Kleinbauer, T., Poller, P., Becker, T., Renals, S., and Kilgour, J. (2009). Extrinsic summarization evaluation: A decision audit task. *TSLP*, 6(2).

Otterbacher, J., Erkan, G., and Radev, D. R. (2005). Using random walks for question-focused sentence retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 915–922, Stroudsburg, PA, USA. Association for Computational Linguistics.

Park, S. and An, D. U. (2010). Automatic query-based personalized summarization that uses pseudo relevance feedback with nmf. In *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication*, ICUIMC '10, pages 61:1–61:7, New York, NY, USA. ACM.

Radev, D., Allison, T., Blair-Goldensohn, S., Blitzer, J., Çelebi, A., Dimitrov, S., Drabek, E., Hakim, A., Lam, W., Liu, D., Otterbacher, J., Qi, H., Saggion, H., Teufel, S., Topper, M., Winkel, A., and Zhang, Z. (2004). MEAD - a platform for multidocument multilingual text summarization. In *LREC 2004*, Lisbon, Portugal.

Rose, D. E., Orr, D., and Kantamneni, R. G. P. (2007). Summary attributes and perceived search quality. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 1201–1202, New York, NY, USA. ACM.

Sakai, T. and Sparck-Jones, K. (2001). Generic summaries for indexing in information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 190–198, New York, NY, USA. ACM.

Sanderson, M. (1998). Accurate user directed summarization from existing tools. In *Proceedings of the seventh international conference on Information and knowledge management*, CIKM '98, pages 45–51, New York, NY, USA. ACM.

Savenkov, D., Braslavski, P., and Lebedev, M. (2011). Search snippet evaluation at yandex: lessons learned and future directions. In *Proceedings of the Second international conference on Multilingual and multimodal information access evaluation*, CLEF'11, pages 14–25, Berlin, Heidelberg. Springer-Verlag.

Tombros, A. and Sanderson, M. (1998). Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 2–10, New York, NY, USA. ACM.

Trappett, M., Geva, S., Trotman, A., Scholer, F., and Sanderson, M. (2012). Overview of the inex 2011 snippet retrieval track. In *Proceedings of the 10th international conference on Initiative for the evaluation of XML retrieval*, INEX'11, Berlin, Heidelberg. Springer-Verlag.

Turpin, A., Scholer, F., Jarvelin, K., Wu, M., and Culpepper, J. S. (2009). Including summaries in system evaluation. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 508–515, New York, NY, USA. ACM.

Varadarajan, R. and Hristidis, V. (2005). Structure-based query-specific document summarization. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 231–232, New York, NY, USA. ACM.

Wang, C., Jing, F., Zhang, L., and Zhang, H.-J. (2007). Learning query-biased web page summarization. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 555–562, New York, NY, USA. ACM.

Wasson, M. (2002). Using summaries in document retrieval. In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4*, AS '02, pages 27–44, Stroudsburg, PA, USA. Association for Computational Linguistics.

White, R. W., Jose, J. M., and Ruthven, I. (2003). A task-oriented study on the influencing effects of query-biased summarisation in web searching. *Inf. Process. Manage.*, 39(5):707–733.

White, R. W., Ruthven, I., and Jose, J. M. (2002). Finding relevant documents using top ranking sentences: an evaluation of two alternative schemes. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 57–64, New York, NY, USA. ACM.

Zheng, Z., Chen, K., Sun, G., and Zha, H. (2007). A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 287–294, New York, NY, USA. ACM.