Generating Informative Snippet to Maximize Item Visibility*

Mahashweta Das[‡], Habibur Rahman[‡], Gautam Das[‡,†], Vagelis Hristidis[‖]
[‡]University of Texas at Arlington; [†]Qatar Computing Research Institute; [‖]University of California, Riverside
[‡]{mahashweta.das@mavs, habibur.rahman@mavs, gdas@cse}.uta.edu, [†]gdas@qf.org.qa, [‖]vagelis@cs.ucr.edu

ABSTRACT

The widespread use and growing popularity of online collaborative content sites has created rich resources for users to consult in order to make purchasing decisions on various items such as e-commerce products, restaurants, etc. Ideally, a user wants to quickly decide whether an item is desirable, from the list of items returned as a result of her search query. This has created new challenges for producers/manufacturers (e.g., Dell) or retailers (e.g., Amazon, eBay) of such items to compose succinct summarizations of web item descriptions, henceforth referred to as *snippets*, that are likely to maximize the items' visibility among users. We exploit the availability of user feedback in collaborative content sites in the form of tags to identify the most important item attributes that must be highlighted in an item snippet. We investigate the problem of finding the top-k *best* snippets for an item that are likely to maximize the probability that the user preference (available in the form of search query) is satisfied. Since a search query returns multiple relevant items, we also study the problem of finding the *best diverse* set of snippets for the items in order to maximize the probability of a user liking at least one of the top items. We develop an exact top-k algorithm for each of the problem and perform detailed experiments on synthetic and real data crawled from the web to to demonstrate the utility of our problems and effectiveness of our solutions.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

collaborative tagging, item attributes, snippet, naive bayes

1. INTRODUCTION

Motivation: The widespread use and growing popularity of online collaborative content sites has created rich resources for users to consult in order to make purchasing decisions on various items such as e-commerce products, travel movies, restaurants, etc. Collaborative content sites (e.g., Amazon, Yelp) contain millions of web items (i.e., items available over the web). For example, the review site Yelp contains more than 25,000 restaurants listed only for New York. Faced with such overwhelming choices, it is becoming increasingly important for the producers/manufacturers (e.g., Dell for laptops) or retailers (e.g., Amazon, eBay) of such items to help a user quickly discover the items she is interested in from the list of items returned as a result of her search query. A popular technique is to associate each item with a short description that provides the *first impression* of the item, i.e., only the necessary and interesting details to help a user make a decision. Such succinct summarizations of web item descriptions are referred to as *snippets*. Typically, an item snippet only involves a fraction of the item attributes. For example, a laptop that highlights the features 2^{nd} `Gen Intel Core i7 processor` and `14″ display, 0.9″ thin, 4lbs weight` in its snippet is likely to influence a prospective customer decision in its favor, especially if she is looking for portable and powerful laptop. However, the snippets shown with items are pre-defined and static. A user searching for a stylish laptop would not benefit from the above snippet.

Various collaborative content sites today encourage users to actively participate by assigning *tags* to online resources with a purpose to promote their contents and to allow users to share, discover and organize them. We exploit the availability of user feedback in the form of tags to *automatically* generate item snippet that are likely to maximize its visibility among users. We perform aggregate analytics over item attributes and tags to identify the *salient features* that are responsible for the positive feedback the item has received so far and that would be highlighted in its snippet. We also diversify the snippets of items returned as a result of a search query in order to maximize the chances of a user liking at least one of the returned items.

Our Problem: There are several challenges associated with finding the best item snippet to be presented to a user engaged in a given context, such as integrating user's past browsing history and behavioral attributes, designing the appropriate mathematical optimization model to maximize the value for users, advertisers, publishers, etc. We focus on the novel aspect of building an item snippet as a succinct summary of its specifications that matches the user's search

*The work of Mahashweta Das, Habibur Rahman, and Gautam Das is partially supported by NSF grants 0812601, 0915834, 1018865, a NHARP grant from the Texas Higher Education Coordinating Board, and grants from Microsoft Research and Nokia Research. The work of Vagelis Hristidis is partially supported by NSF grants IIS-1216032 and IIS-1216007.
*Part of this work was done while Habibur Rahman was interning at Qatar Computing Research Institute.

CIKM '13 Burlingame, CA USA
Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...$15.00.
.

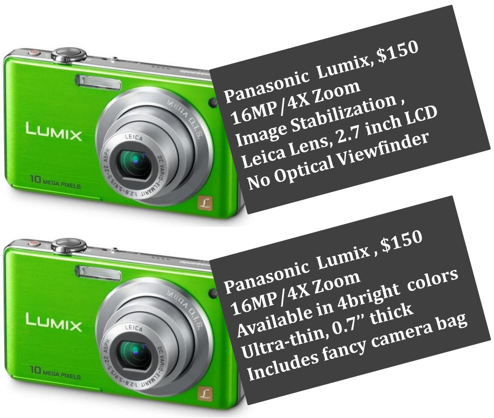


Figure 1: Top: Pre-defined and static snippet for camera, Bottom: Informative snippet for camera for query stylish digital camera

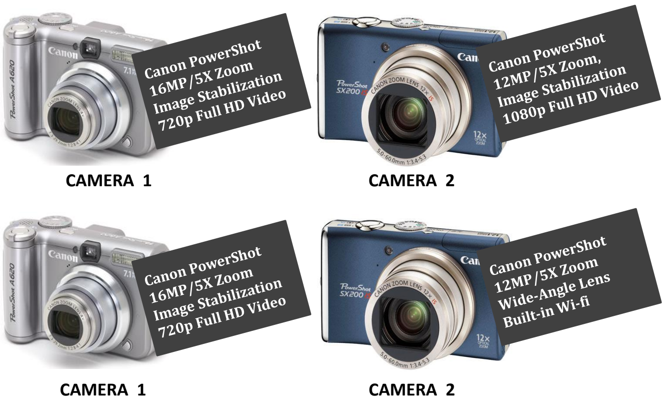


Figure 2: Top: Informative snippets, Bottom: Diversified Informative snippets for Cameras 1 and 2 returned for query travel-purpose digital camera

query and highlights the features, that were responsible for the positive feedback left by the past users with similar preferences. For example, if a user is looking for an adventurous and budget-friendly Europe backpacking trip package, a returned trip snippet must highlight the relevant related features `youth-hostel, Eurail Youth Pass` and `free city-attractions` to draw his attention. Intuitively, we discover in the database of trips, those attributes that are responsible for the trip receiving the tags adventurous and budget-friendly by past users. We refer to this problem as the **Informative Snippet Design (ISD) Problem** for a single item, that identifies the salient item attributes to be highlighted in its snippet in order to maximize the probability of the user preference (available in the form of search query) being satisfied. This can be extended to the top-k version where we return the top-k snippets, which can be post-processed by the manufacturers, retailers, etc. to accommodate the more traditional factors. We also envision the utility of dynamic snippet, as opposed to regular pre-defined and static summaries (e.g., in faceted navigation of Amazon and eBay) to match a user's search query effectively. For example, a user looking for stylish digital camera would benefit more from the snippet in the bottom row of Figure 1 than the pre-defined and static description in the top row of Figure 1.

The ISD problem intends to identify the *relevance* of a snippet to a user search query. In this work, we consider the following three categories of conjunctive queries:
(i) General-purpose queries - User search queries that do not express specific user preferences (e.g., laptop, digital camera, cellphone, etc.). For this type of queries, the ISD goal is to maximize the probability of an item snippet receiving all positive tags that existing items have received in the past.
(ii) Tag-driven queries - User search queries that express a user's preference by short keywords or tags (e.g., lightweight laptop, modern cellphone). For this type of queries, the ISD goal is to maximize the probability of an item snippet receiving the user-specified tags (and its synonyms).
(iii) Attribute-driven queries - User search queries that express a user's preference by attribute values (e.g., `SLR` camera, `3g` cellphone). For this type of queries, the ISD goal is to maximize the probability of an item snippet receiving all positive tags that existing items (in the same category) with similar attribute values have received so far.

Henceforth, we refer to the set of tags associated with the user search query as the *desirable* tags. Note that, in addition to an item's technical specifications, several other implicit factors such as item quality and utility, user behavior, etc. influence tagging behavior. We refer to related literature [2] and choose to focus on content-based tagging feedback to identify the salient features of an item.

The list of items returned as a result of a user search query are often very similar to each other, and hence would have similar snippets generated. Therefore, it is necessary to *diversify* the snippets associated with the returned items in order to increase the chances of a user liking any one of the top returned items. In this paper, we study the problem of **Diversified Informative Snippets Design (DISD) Problem** for a list of items returned by a search query, to find snippets that highlight the most relevant and the most diverse features. For example, a user looking for travel-purpose digital camera would benefit more from the diversified snippets in the bottom row of Figure 2 for Cameras 1 and 2, than the snippets in the top row of Figure 2. Extracting a set of diverse features, that covers the various aspects of the underlying dataset, is a problem of automated facet discovery which is known to improve user experience in the web. While faceted search employed by sites (e.g., Amazon, eBay) performs pre-defined top-down navigation on the concept hierarchy, where all features of the currently selected concept are displayed, our objective is to highlight the important features as well as diverse features. Diversification of search has been studied in recent times in several contexts with many different approaches, majority of which focuses on a scoring function that takes both query relevance and diversity into consideration [1][11]. We measure diversity as a function of exclusivity and coverage of attributes in the snippets of items, while ensuring that the snippet selected for each of the items has a relevance score close to the best possible snippet score for that item.

Technical Challenges and Solutions: Solving the informative snippet design problem is technically challenging. Complex dependencies exist among tags and item attributes. Additionally, the task of finding the best set of attributes maximizing the probability of an item snippet receiving all desirable tags requires us to exhaustively evaluate an exponential number of combinations. In this paper, we consider the very popular Naive Bayes Classifier with the simplistic conditional independence assumption for tag and attribute modeling because of its success in [2]. We introduce the idea of *composite tag*, a single tag representing all the desirable tags that alleviates the computational challenges associated with finding the best snippet for an item. We propose an exact top-k algorithm that performs significantly better than the naive brute-force algorithm for the ISD problem. Our DISD problem is conspicuously different from diversity aware search: diversity aware search aims to find the top-k relevant items from the set of all n relevant items returned as a result of search query; the DISD problem aims to find a result (i.e., snippet) for each of the n relevant items, where each item has a set of top-k results to choose from. We develop a novel exact top-k algorithm for the DISD problem based on non-trivial adaptations of top-k query processing techniques in [9][11]. We experiment with both synthetic and real data crawled from the web to demonstrate the effectiveness of our algorithms and conduct user studies to validate that our snippets are useful to draw user attention.

In summary, we make the following main contributions:

- We introduce the ISD problem of designing the top-k snippets for an item by leveraging available user feedback in the form of tags, in order to maximize the item's visibility among users. Since a search query returns multiple relevant items, we also introduce the DISD problem of finding the best diverse set of snippets for multiple items in order to maximize the chances of a user liking at least one of the returned items.
- We develop for each problem, an exact top-k algorithm that works well in practice.
- We perform detailed experiments on synthetic and real data crawled from the web to demonstrate the utility of our problems and effectiveness of our solutions.

2. PROBLEM FRAMEWORK

Let $\mathbb{D} = \{o_1, o_2, ..., o_N\}$ be a collection of N items, where each item entry is defined over the attribute set $A = \{A_1, A_2, ..., A_m\}$ and the tag dictionary space $\mathbb{T} = \{T_1, T_2, ..., T_r\}$. Each attribute A_i can take one of several values a_i from a multi-valued categorical domain D_i, or one of two values $\{0, 1\}$ if a boolean dataset is considered. A tag T_j is a bit where a 0 implies the absence of a tag and a 1 implies the presence of a tag for item o. Each item is thus a vector of size $(m + r)$, where the first m positions correspond to a vector of attribute values, and the remaining r positions correspond to a boolean vector.

Consider a query which picks a set of desirable tags $T^d = \{T_1, ..., T_z\} \subseteq \mathbb{T}$. The objective of the ISD problem is to determine s of m attributes for building the snippet S_o of an item o, such that the probability of attracting all desirable tags $T_j \in T^d$ is maximized. The top-k snippets of item o are represented as $S_o^1, S_o^2, \ldots, S_o^k$.

Given a training set as the dataset described above, we build Naive Bayes Classifier (NBC), that classify tags given attributes (one classifier per tag) defines the probability that a snippet S_o is annotated by tag T_j. If $\{a_1, a_2, ..., a_s\}$ are the attribute values in S_o, the classifier for tag T_j defines the probability that snippet S_o of item o draws tag T_j, as:

$$\begin{aligned} Pr(T_j \mid S_o) &= Pr(T_j \mid a_1, a_2, ..., a_s) \\ &= \frac{Pr(T_j).\Pi_{i=1}^{s} Pr(a_i \mid T_j)}{Pr(a_1, a_2, ..., a_s)} \end{aligned} \quad (1)$$

$$Pr(T_j{}' \mid S_o) = \frac{Pr(T_j{}').\Pi_{i=1}^{s} Pr(a_i \mid T_j{}')}{Pr(a_1, a_2, ..., a_s)} \quad (2)$$

Since $Pr(T_j \mid S_o) + Pr(T_j{}' \mid S_o) = 1$, from Equations 1, 2:

$$\begin{aligned} Pr(a_1, a_2, ..., a_s) &= Pr(T_j).\Pi_{i=1}^{s} Pr(a_i \mid T_j) + \\ &\quad Pr(T_j{}').\Pi_{i=1}^{s} Pr(a_i \mid T_j{}') \end{aligned} \quad (3)$$

From Equations 1, 3:

$$\begin{aligned} Pr(T_j \mid S_o) &= Pr(T_j \mid a_1, a_2, ..., a_s) \\ &= \frac{Pr(T_j).\Pi_{i=1}^{s} Pr(a_i \mid T_j)}{Pr(T_j).\Pi_{i=1}^{s} Pr(a_i \mid T_j) + Pr(T_j{}').\Pi_{i=1}^{m} Pr(a_i \mid T_j{}')} \\ &= \frac{1}{1 + \frac{Pr(T_j{}')}{Pr(T_j)} \Pi_{i=1}^{s} \frac{Pr(a_i|T_j{}')}{Pr(a_i|T_j)}} \end{aligned} \quad (4)$$

The probability of snippet S_o of an item o drawing all desirable tags $T^d = \{T_1, ..., T_z\}$, i.e., the relevance score is:

$$\begin{aligned} f(S_o, T^d) &= Pr(T_1, ..., T_z \mid S_o) \\ &= Pr(T_1 \mid S_o)....Pr(T_z \mid S_o) \\ &= \Pi_{j=1}^{z} \frac{1}{1 + \frac{Pr(T_j{}')}{Pr(T_j)} \Pi_{i=1}^{s} \frac{Pr(a_i|T_j{}')}{Pr(a_i|T_j)}} \end{aligned} \quad (5)$$

Note that the task of finding the best snippet that maximizes Equation 5 is difficult, even for $k = 1$ [2]. Hence, we introduce the idea of composite tag.

Composite Tag: A composite tag is a single tag T that consists of the collection of desirable tags in T^d, and alters the ISD relevance score computation function to:

$$\begin{aligned} f(S_o, T) &= f(S_o, T^d) \\ &= Pr(T_1, ..., T_z \mid S_o) \\ &= Pr(T \mid S_o) \\ &= \frac{1}{1 + \frac{Pr(T')}{Pr(T)} \Pi_{i=1}^{s} \frac{Pr(a_i|T')}{Pr(a_i|T)}} \end{aligned} \quad (6)$$

The consideration of composite tag reduces the computational complexity of maximizing the *sum-of-product* quantity in Equation 5. The scoring function now intends to maximize a product quantity of the form $\Pi_{i=1}^{s} \frac{Pr(a_i|T)}{Pr(a_i|T')}$ in Equation 6.

If there are sufficient instances in the training dataset that have $T = T^d = \{T_1, ..., T_z\} \subseteq \mathbb{T}$, we can directly compute probabilities of the form $Pr(a_i \mid T), Pr(a_i \mid T')$, $\forall i = 1 \ldots m$. If the number of instances is insufficient, we compute the probabilities by considering conditional independence in the following way:

$$\begin{aligned} Pr(a_i \mid T) &= Pr(T \mid a_i).\frac{Pr(a_i)}{Pr(T)} \\ &= Pr(T_1, T_2, \ldots, T_z \mid a_i).\frac{Pr(a_i)}{Pr(T)} \\ &= Pr(T_1 \mid a_i).Pr(T_2 \mid a_i) \ldots Pr(T_z \mid a_i).\frac{Pr(a_i)}{Pr(T)} \end{aligned}$$

Quantities of the form $Pr(a_i \mid T')$ are difficult to resolve since they cannot be reduced using the conditional independence assumption. However, since $Pr(T)$ is small in this case, $Pr(T')$ is large ≈ 1. Therefore, we approximately estimate $Pr(a_i \mid T')$ by computing $Pr(a_i \mid \mathbb{D})$.

We are now ready to formally define our problems.

INFORMATIVE SNIPPET DESIGN (ISD) PROBLEM: *Given a user search query expressed as a composite tag T (i.e., the set of desirable tags T^d) and an item o from a dataset of tagged items $\mathbb{D} = \{o_1, o_2, ..., o_N\}$, design k snippets $S_o^1, S_o^2, \ldots, S_o^k$ of size s for o that have the highest score of receiving all desirable tags, given by Equation 6.*

DIVERSIFIED INFORMATIVE SNIPPET DESIGN (DISD) PROBLEM: *Given a list of n items* $\{o_1, o_2, \ldots, o_n\}$ *returned by the search engine from a dataset* $\mathbb{D}$ *of* N *items for user query* T*, and the top-k snippets* $\{\{S^1_{o_1}, S^2_{o_1}, \ldots, S^k_{o_1}\}$, $\{S^1_{o_2}, S^2_{o_2}, \ldots, S^k_{o_2}\}$, ..., $\{S^1_{o_n}, S^2_{o_n}, \ldots, S^k_{o_n}\}\}$ *of size s for each of the n items, determine n snippets* S_{o_1}, S_{o_2}, $\ldots$, S_{o_n} *for the n items respectively such that:*

- $S_{o_i} \epsilon \ \{S^1_{o_i}, S^2_{o_i}, \ldots, S^k_{o_1}\}$, $\forall i = 1 \ldots n$
- *diversity*(S_{o_x}, S_{o_y}) $\geq \tau$, $x \neq y$, $\forall(o_x, o_y), \in \{o_1, .., o_n\}$
- $f(S^1_{o_i}, T) - f(S_{o_i}, T) \leq \theta$, $\forall i = 1 \ldots n$
- *sum* ($f(S_{o_1}, T)$, $\ldots$, $f(S_{o_n}, T)$) *is maximized*

where diversity(S_{o_x}, S_{o_y}) measures the diversity between two snippets S_{o_x} and S_{o_y}; τ is the threshold ensuring the snippets are diverse enough, and θ is the threshold ensuring that the relevance score of a selected item snippet is not far from the relevance score of the best snippet for that item.

Complexity Analysis: The ISD problem scoring function in Equation 6 involves the quantity $\Pi_{i=1}^{s} \frac{Pr(a_i|T')}{Pr(a_i|T)}$, which can be expressed as a sum, $\Sigma_{i=1}^{s} \log[Pr(a_i \mid T')/Pr(a_i \mid T)]$. Therefore, the problem can be formulated as an s-SUM Problem, which is a parameterized version of the well known combinatorial optimization problem SUBSET SUM. The s-SUM problem is fundamentally connected to several NP-hard problems and is proved to be W[1]-hard [4]. The DISD Problem objective is to identify the best combination of snippets for n items, where each item has a set of top-k snippets to choose from, such that the total score (i.e., relevance to query) of chosen snippets is maximized, subject to diversity constraints being satisfied. This problem can be expressed as the FACILITY DISPERSION PROBLEM in computational geometry literature, where the task is choose p out of n facilities, so as to maximize some function of the distances between facilities. Our DISD problem can be formulated as the MAXSUMDISPERSION problem. Both our problems are NP-Complete by reduction from SUBSET SUM and SET COVER respectively, the proofs of which are skipped because of space constraints.

3. ISD ALGORITHM

In this section, we propose an efficient algorithm for solving the Informative Snippet Design (ISD) problem.

A brute-force exhaustive approach (henceforth, referred to as **Naive-ISD**) to solve the problem requires us to design all ${}^{m}C_s$ possible snippets S^1_o, S^2_o, $\ldots$, $S_o^{{}^{m}C_s}$ for item o and composite tag T, and compute $f(S_o, T)$ for each possible snippet in order to identify the top-k snippets. If the snippet size s and the total number of attributes m are small, Naive-ISD is capable of returning the top-k results in reasonable amount of time. However, since m and s are usually large in real data, we develop an efficient and practical algorithm.

Our proposed algorithm is an exact top-k technique, Exact-ISD (**E-ISD**) based on an interesting adaptation of Fagin's Threshold Algorithm (TA) [5]. We create s identical lists $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_s\}$ for identifying snippets $\{S^1_o, S^2_o, \ldots, S^k_o\}$ of size s for item o where each list $\mathcal{L}_i$ contains m values of the form $\sqrt[s]{\frac{Pr(T)}{Pr(T')}} \cdot \frac{Pr(a_i|T)}{Pr(a_i|T')}$ corresponding to the m attributes in descending order of magnitude. The sorting is done on the contributions made by attributes to maximize the scoring function in Equation 6. The lists are accessed in round robin fashion and for every combination of attributes from the lists, we join them to build a snippet. A join is considered to be *valid* if the number of distinct attributes in the join is equal to s and the join combination (without considering the order of attributes participating in join) is not already included in the result set. The complete score of the valid join (i.e., the snippet) is resolved by Equation 6. We maintain a buffer of size k, called $top - k$ buffer, in order to store the k best snippets $\{\{S^1_o, S^2_o, \ldots, S^k_o\}$ for item o. A snippet is stored in the $top - k$ buffer if its score is higher than the MPFS (Maximum Possible Future Score) at a point, which is the upper bound on the score of an unseen snippet. MPFS is computed using the currently indexed entry of a list and top $(s-1)$ entries of any one of the lists, since they are identical.

$$MPFS \;\; = \;\; \frac{1}{1 + (c \cdot h_1 \cdot h_2 \cdot \cdots \cdot h_{s-1})} \qquad (7)$$

where, c is the score of the currently indexed entry and h_1 to h_{s-1} are the scores of the top $(s-1)$ entries from any list.

4. DISD ALGORITHM

In this section, we propose an efficient algorithm for solving the Diversified Informative Snippet Design (DISD) problem.

The objective of DISD is to diversify the snippets of items returned as a result of search query in order to maximize the chances of a user liking at least one of the top items. Similar to related research on diversity aware search, we intend to determine the item snippets based on both their relevance to the search query as well as their dissimilarity to the other selected snippets. We emphasize word sense diversification in the snippets for diversity and measure diversity as categorical distance, based on the Hamming metric.

Diversity: Given attribute set $A = \{A_1, A_2, \ldots, A_m\}$ where each attribute A_i can take one of several values a_i from a multi-valued categorical domain D_i, or one of two values {0, 1} if a boolean dataset is considered, we build feature (i.e., description) vectors $\vec{d}$ of length $n_d = \Sigma_{i=1}^{m}|D_i|$, where values in $\vec{d}$ are set to 1 or 0 depending on the snippet under consideration. The diversity between snippets S_{o_x} and S_{o_y} of items o_x and o_y having description vectors d_{o_x} and d_{o_y} is:

$$diversity(S_{o_x}, S_{o_y}) = \Sigma_{j=1}^{n_d}(d_{o_x}[j] \neq d_{o_y}[j]) \qquad (8)$$

where j is the vector index and $d_{o_x}[j] \neq d_{o_y}[j]$ is 1 if they are different; otherwise 0. In this study, it is not our goal to advocate one particular diversity measure over another. Rather, we focus on formalizing the problem and developing efficient solutions. The relevance score of a snippet is computed by Equation 6.

A brute-force exhaustive approach (henceforth, referred to as **Naive-DISD**) to solve the problem requires us to explore k^n combinations, and compute sum $\Sigma_{i=1}^{n}(f(S_{o_i}, T)$,subject to diversity($S_{o_x}$, S_{o_y}) $\geq \tau$, for all pairs of o_x and o_y. Thus, we develop an efficient and practical algorithm Exact-DISD (**E-DISD**) based on interesting and non-trivial adaptations top-k querying techniques in [11][9].

We create n lists $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n\}$ corresponding to the n items returned by search engine for a user query. Each list L_i contains the top-k snippets $\{S^1_{o_i}, S^2_{o_i}, \ldots, S^k_{o_i}\}$ for item o_i having scores (given by Equation 6) sorted in decreasing order of score. Note that, for each list we only include the snippets that have relevance score difference up to θ from the top one for that list (i.e., item). The lists are accessed in round robin fashion and for every join, we check if it is a valid join. A join is considered to be *valid* if the diversity constraint is satisfied, i.e., any two snippets S_{o_x} and S_{o_y} for items o_x and o_y ($S_{o_x} \in \{S^1_{o_x}, S^2_{o_x}, \ldots, S^k_{o_x}\}$, similarly for S_{o_y}) are dissimilar and have $diversity(S_{o_x}, S_{o_y})$ exceeding a user-provided threshold, τ. The complete score of the join is resolved by summing over the snippet scores, i.e., $\Sigma^n_{i=1} f(S^j_{o_i}, T)$, $j \in \{1, k\}$. Our objective is to identify the top-1 combination which would return the n snippets S_{o_1}, $S_{o_2}, \ldots, S_{o_n}$ for the n items, where $S_{o_i} \epsilon \{S^1_{o_i}, S^2_{o_i}, \ldots, S^k_{o_1}\}$, $\forall i = 1 \ldots n$. The top-1 result is returned if its score is higher than the MPFS (Maximum Possible Future Score), which is the upper bound on the score of an unseen combination. To compute MPFS, we assume that the current entry from a list is joined with the top entries from all other lists, as given below:

$$MPFS = max((c_1 + h_2 \cdots + h_n), \ldots, (h_1 + h_2 \cdots + c_n)) \quad (9)$$

where, where c_i and h_i are the last seen and top entries from list $\mathcal{L}_i$ respectively.

5. EXPERIMENTS

We conduct a set of comprehensive experiments using both synthetic and real datasets for quantitative (Section 5.1) and qualitative analysis (Section 5.2) of our proposed algorithms. Our quantitative performance indicator is *efficiency* of the algorithms, measured by running time. We also conduct a detailed use-case evaluation, where we show how our snippets are helpful to draw user attention.

System configuration: Our prototype system is implemented using C#. All experiments were conducted on an Windows 7 machine with 2.30Ghz Intel i5 processor, 64 bit Operating System and 6GB RAM.

Real Car Dataset: We crawl a real dataset of 606 cars[1] spanning 34 brands from Yahoo! Autos[2] for the year 2010. The products contain technical specifications as well as ratings and reviews, which include pros and cons. We parse a total of 60 attributes: 25 numeric, and 35 boolean and categorical (which we generalize to boolean). The total number of reviews we extract is 2,180. We process the text listed under pros in each review to identify a set of 15 desirable tags such as `fuel economy`, `stylish exterior`, etc, using the keyword extraction toolkit AlchemyAPI[3].

Synthetic Dataset: We generate a large boolean matrix of dimension 10,000 (items)×100 (50 attributes + 50 tags) and randomly choose submatrices of varying sizes, based on our experimental setting. We split the 50 independent and identically distributed attributes into four groups, where the value is set to 1 with probabilities of 0.75, 0.15, 0.10 and 0.05 respectively. For each of the 50 tags, we pre-define relations by randomly picking a set of attributes that are correlated to it. A tag is set to 1 with a probability p if majority of the attributes in its pre-defined relation have boolean value 1.

We use the synthetic datasets for quantitative experiments, while the real dataset is used in the qualitative study.

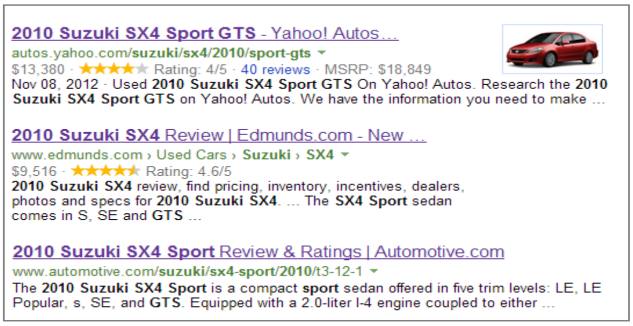
5.1 Quantitative Results: Performance

Efficiency: We first compare the performance behavior of the ISD algorithms - Naive-ISD and E-ISD, in Figures 3 and 4. Since Naive-ISD can only work for small problem instances, we pick a subset from the synthetic dataset containing 1000 items, 50 attributes, 10 tags. Figure 3 compares the execution time of the ISD algorithms for 1000 items, 10 tags, having snippet size $s = 5$ and top-k's $k = 5$, with varying number of attributes, m. We observe that our E-ISD outperforms the Naive-ISD method as the number of attributes increases. Next, we analyze the time taken when the snippet size s varies in Figure 4. We consider a synthetic data containing 1000 items, 20 attributes, 10 tags, $k = 10$, and vary s from 4 to 16. We observe that the time taken by Naive-ISD is again much more than that taken by E-ISD. Note that the time taken by E-ISD is affected by s, since the number of lists considered is equal to the snippet size.

Figure 5 compares the execution time of our DISD algorithms - Naive-DISD and E-DISD. Recall that, the computational complexity of the DISD problem is dependent on the number of relevant items n. Therefore, we evaluate the performance behavior of our DISD algorithm by varying n. Figures 5 shows how the execution time of both Naive-DISD and E-DISD rises with increase in the number of relevant items n, when synthetic data of 1000 items, 30 attributes, 10 tags, snippet size $s = 10$, $k = 10$, and $\tau = 2$ is considered.

5.2 Qualitative Results: Performance

We use the real cars dataset to validate that our algorithms draw interesting snippets highlighting the desirability of car specifications (i.e., attribute values), as opposed to the general snippets that are currently returned by the search engines. For a user looking for a used japanese sports car, one of the top cars returned by the search engine is "Suzuki SX4 Sport". For the 2010 Suzuki SX4 Sport GTS, the usual snippets displayed by the search engine and/or the retail sites are shown in Figure 6. As we see, the snippet compositions are not striking and mention the usual high-level car specifications, that other cars returned by the search query (or a different search query about cars) would mostly display.



2010 Suzuki SX4 Sport GTS - Yahoo! Autos ...
autos.yahoo.com/**suzuki/sx4/2010/sport-gts**
$13,380 · ★★★★☆ Rating: 4/5 · 40 reviews · MSRP: $18,849
Nov 08, 2012 · Used **2010 Suzuki SX4 Sport GTS** On Yahoo! Autos. Research the **2010 Suzuki SX4 Sport GTS** on Yahoo! Autos. We have the information you need to make ...

2010 Suzuki SX4 Review | Edmunds.com - New ...
www.edmunds.com › Used Cars › **Suzuki** › **SX4**
$9,516 · ★★★★★ Rating: 4.6/5
2010 Suzuki SX4 review, find pricing, inventory, incentives, dealers, photos and specs for **2010 Suzuki SX4**. ... The **SX4 Sport** sedan comes in S, SE and **GTS** ...

2010 Suzuki SX4 Sport Review & Ratings | Automotive.com
www.automotive.com/**suzuki/sx4-sport/2010**/t3-12-1
The **2010 Suzuki SX4 Sport** is a compact **sport** sedan offered in five trim levels: LE, LE Popular, s, SE, and **GTS**. Equipped with a 2.0-liter I-4 engine coupled to either ...

Figure 6: Currently available snippets for 2010 Suzuki SX4 Sport GTS

However, our E-ISD algorithm returns the following relevant snippet, highlighting salient and query-relevant attributes like `mileage`, `horsepower`, `safety features`, etc, thereby confirming the utility of our problem and effectiveness of our solution.

2010 Suzuki SX4 GTS: $13,380 based on 24,000 driven miles; 23 mpg city / 30 mpg hwy; 148 hp; MPFI Engine; KYB(R) Shock Absorbers and Sport Ride Type

[1] Recall that, the number of items in the dataset is not important for the execution cost.

[2] http://autos.yahoo.com/

[3] http://www.alchemyapi.com/

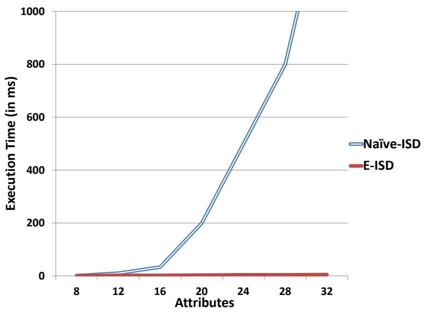


Figure 3: ISD: Execution time for varying m (Synthetic data)

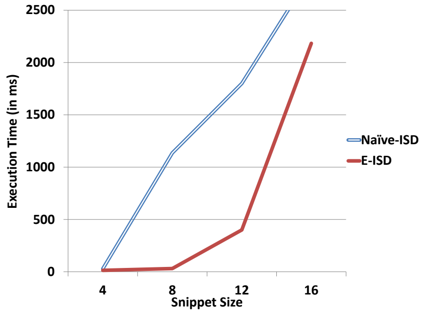


Figure 4: ISD: Execution time for varying s (Synthetic data)

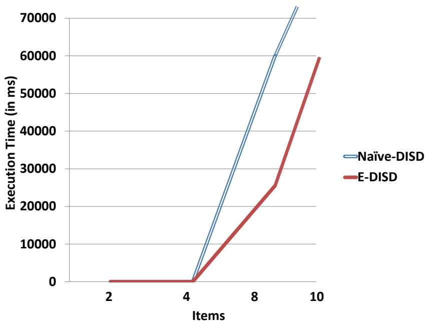


Figure 5: DISD: Execution time for varying n (Synthetic data)

Next, we study how diverse snippets are returned by our A-DISD algorithm. For a user looking for a `used audi a4`, suppose the top cars returned by the search engine include "FrontTrak Multitronic", "Quattro Manual", "Quattro Tiptronic", and "Avant Quattro Tiptronic". The cars share several attributes in common, and are hence likely to generate similar snippets. However, our A-DISD algorithm identifies several unique features that these used cars have such as `front fog-light` in FrontTrak Multitronic, `first-aid kit` in Quattro Manual, `garage door-opener` in Quattro Tiptronic, and `delayed courtesy light` in Avant Quattro Tiptronic. Such diverse snippets returned to a user looking for an used Audi A4 are likely to increase the chances of the user clicking on one of them.

6. RELATED WORK

Web Advertisement and Snippets: There has been a lot of work on web advertisement and snippet construction [13], most of which leverages text mining and natural language processing techniques to identify the top sentences to display [12] or in response to user search query [8]. There are several research challenges associated with finding the best ad [10].We consider the novel task of snippet generation by leveraging collaborative tagging feedback.

Collaborative Tagging Mining: The dynamics of social tagging has been an active research area in recent years, with several papers focusing on leveraging collaborative tagging feedback for improving recommendation [3], designing new products [2], etc. Several paper focuses on the task of tag prediction, with [7] using Naive Bayes for tag prediction.

Techniques in our Work: Our top-k algorithms in the paper are inspired by the rich body of work in [5] [9] [11]. We propose approximation algorithms which borrows ideas from popular combinatorial optimization problems in the literature [6]. Finally, our snippet diversification semantics is based on existing work [1][9] that support diversity on search results, though our technical objective is conspicuously different from diversifying search problems.

7. CONCLUSION

We study the novel problem of leveraging collaborative tagging for generating informative snippets to maximize its visibility among users. We formally define two problems - Informative Snippet Design problem for a single item, and Diversified Informative Snippets Design problem for a set of items, and develop exact top-k algorithms that are experimentally shown to work well in practice. However, since both our algorithms have exponential complexity in the worst case, we intend to develop approximation algorithms with theoretical bounds in the future. We also intend to evaluate the applicability of our framework for generating snippets of non-commercial contents such as blogs, musical pieces, etc.

8. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, pages 5–14, 2009.

[2] M. Das, G. Das, and V. Hristidis. Leveraging collaborative tagging for web item design. In *KDD*, pages 538–546, 2011.

[3] M. Das, G. D. F. Morales, A. Gionis, and I. Weber. Learning to question: Leveraging user preferences for shopping advice. In *KDD*, 2013.

[4] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness ii: On completeness for w[1]. *Theoretical Computer Science*, 141(1&2):109–131, 1995.

[5] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. *J. Comput. Syst. Sci.*

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[7] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *SIGIR*, pages 531–538, 2008.

[8] Y. Huang, Z. Liu, and Y. Chen. Query biased snippet generation in xml search. In *SIGMOD Conference*, pages 315–326, 2008.

[9] I. F. Ilyas, D. Martinenghi, and M. Tagliasacchi. Rank-join algorithms for search computing. In *SeCO Workshop*, pages 211–224, 2009.

[10] A. Kashyap and V. Hristidis. Comprehension-based result snippets. In *CIKM*, pages 1075–1084, 2012.

[11] L. Qin, J. X. Yu, and L. Chang. Diversifying top-k results. *PVLDB*, 5(11):1124–1135, 2012.

[12] A. Turpin, Y. Tsegay, D. Hawking, and H. E. Williams. Fast generation of result snippets in web search. In *SIGIR*, pages 127–134, 2007.

[13] R. White, I. Ruthven, and J. M. Jose. Finding relevant documents using top ranking sentences: an evaluation of two alternative schemes. In *SIGIR*, pages 57–64, 2002.