Using Micro-Reviews to Select an Efficient Set of Reviews

Thanh-Son Nguyen*
Information Systems Dept.
VNU University of Engineering and Technology, Vietnam
ant.sonnt@gmail.com

Hady W. Lauw
School of Information Systems
Singapore Management University
hadywlauw@smu.edu.sg

Panayiotis Tsaparas
Dept. of Computer Science
University of Ioannina
Greece
tsap@cs.uoi.gr

ABSTRACT

Online reviews are an invaluable resource for web users trying to make decisions regarding products or services. However, the abundance of review content, as well as the unstructured, lengthy, and verbose nature of reviews make it hard for users to locate the appropriate reviews, and distill the useful information. With the recent growth of social networking and micro-blogging services, we observe the emergence of a new type of online review content, consisting of bite-sized, 140 character-long reviews often posted reactively on the spot via mobile devices. These *micro-reviews* are short, concise, and focused, nicely complementing the lengthy, elaborate, and verbose nature of full-text reviews.

We propose a novel methodology that brings together these two diverse types of review content, to obtain something that is more than the sum of its parts. We use micro-reviews as a crowdsourced way to extract the salient aspects of the reviewed item, and propose a new formulation of the review selection problem that aims to find a small set of reviews that *efficiently* cover the micro-reviews. Our approach consists of a two-step process: matching review sentences to micro-reviews and then selecting reviews such that we cover as many micro-reviews as possible, with few sentences. We perform a detailed evaluation of all the steps of our methodology using data collected from Foursquare and Yelp.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms, Experimentation

Keywords

Micro-review, review selection

*Work done while visiting Living Analytics Research Center, Singapore Management University.

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505568.

1. INTRODUCTION

Online reviews are pervasive. Today, for almost any product or service, we can find ample review content in various Web sources. For instance, Amazon.com hosts product reviews as part of an online shopping experience to assist their customers in determining which product is most suitable for their need. Yelp.com is a popular site for restaurant reviews, assisting diners to plan restaurant visits. Reviews are immensely useful in aiding decision-making, because they allow the readers to anticipate what their experience would potentially be based on the prior experiences of others, without having to make a trip to the store or the restaurant.

While useful, the deluge of online reviews also causes some issues. Readers are inundated by the numerous reviews, and it is not clear which reviews are worthy of a reader's attention. This is worsened by the length and verbosity of many reviews, whose content may not be wholly relevant to the product or service being reviewed. Reviewers often diverge, meandering around personal details that do not offer any insight about the place being reviewed. Furthermore, it is getting increasingly more difficult to determine the authenticity of a review, whether it has been written by a genuine customer sharing her experience, or by a spammer seeking to mislead[1]. Identifying and selecting high quality reviews to show to the users is a hard task, and it has been the focus of substantial amount of research [3, 7, 12, 10, 24, 9].

With the recent growth of social networking and micro-blogging services, we observe the emergence of a new type of online review content. This new type of content, which we term *micro-reviews*, can be found in micro-blogging services that allow users to "check-in", indicating their current location or activity. For example, at Foursquare, users check in at local venues, such as restaurants, bars, coffee shops. At GetGlue.com, users check in to TV shows, movies, or sports events. Check-ins are also possible within social networking sites such as Facebook, or Twitter. After checking in, a user may choose to leave a 140 character-long message about their experience, effectively a *micro-review* of the place or the activity. Following the Foursquare terminology, we will refer to these messages as *tips*. In the case of restaurants, these tips are frequently recommendations (e.g., what to order) or opinions (what is great or not). For example, this is a Foursquare tip for a popular restaurant in New York: *"Be patient. It's worth the wait. Their ramen has crack in it."*.

Micro-reviews serve as an alternative source of content to reviews for readers interested in finding information about

[1] `http://www.businessweek.com/magazine/a-lie-detector-test-for-online-reviewers-09292011.html`

a place. They have several advantages. *First*, due to the length restriction, micro-reviews are *concise and distilled*, identifying the most salient or pertinent points about the place according to the author. For example, the tip above focuses on the long wait and the quality of the ramen. *Second*, because some micro-reviews are written on site and in the moment right after checking in, they are *spontaneous*, expressing the author's visceral reaction to her experience. This is in contrast to the relatively more contemplative and reflective nature of most reviews, which might express the delayed afterthought of the author. *Third*, because most authors check in by mobile apps, it is likely that these authors are actually at the place when they leave the tips, which makes the tips more likely to be *authentic*. Micro-blogging sites also have the ability, if necessary, to filter out tips without an accompanying check-in, thus, boosting the authenticity of the tips.

Micro-reviews and reviews nicely complement each other. While reviews are lengthy and verbose, tips are short and concise, focusing on specific aspects of an item. At the same time, these aspects cannot be properly explored within 140 characters. This is accomplished in full-blown reviews which elaborate and contemplate on the intricacies of a specific characteristic. Marrying these two different reviewing approaches can yield something greater than the sum of their parts: detailed reviews that focus on aspects of a venue that are of true importance to users. This is the goal of this work.

We consider the following problem. Given a collection of reviews, and a collection of tips about an item, we want to select a small number of reviews that best *cover* the content of the tips. The problem of review selection has been studied in the past [10, 24, 9]. The idea underlying all prior work is to select a small comprehensive set of reviews that carry the most information about an item. In all prior work this is modeled as a *coverage problem*, where the selected reviews are required to cover the different aspects of the item (e.g., product attributes), and the polarity of opinions about the item (positive and negative). To extract the aspects covered by a review, and the sentiment polarity, off-the-shelf tools are usually applied, which rely on supervised techniques trained on manually collected data. Such approaches, although generally successful cannot generalize to arbitrary domains and capture the different aspects that users are interested in, or the different ways to describe them in natural language. Unsupervised techniques such as topic modeling have also been applied (e.g., [14]), however they suffer from the broadness of the topic definition.

We view tips as a crowdsourcing way to obtain the aspects of an item that the users care about, as well as the sentiment of the users. By covering the tips, we effectively identify the review content that is important, and the aspects of the item upon which the reviews need to expand and elaborate. In our formulation, which we outline below, we make sure that the selected reviews are compact, that is, the content does not diverge from what is important about the reviewed item. We view this as an important constraint. Reviews, especially for restaurants or other venues, are often read on mobile devices, where screens are small, and time is short. It is thus important to convey the necessary information as *efficiently* as possible.

Our review selection serves an additional purpose beyond identifying the best reviews to show to a user. It provides a summary of the content of the tips. Tips are short and focused, which is good for quickly zooming in on what is interesting about an item. However, this same property makes it hard to go through a large collection of the tips, since they are disjoint, fragmented and repetitive. On the other hand, full-text reviews make for a much more interesting reading, since there is enough space and time to eloquently describe the item that is being reviewed. By selecting the reviews that cover the tips, we effectively obtain a readable, flowing text that summarizes and expands upon the tip content.

Overview of our approach. We now give a high-level overview of our approach. Given an entity (e.g., a restaurant), we assume we are given as input a collection of reviews $\mathcal{R}$ and a collection of tips $\mathcal{T}$ about the entity. Our goal is to select a subset of reviews $\mathcal{S} \subseteq \mathcal{R}$ that covers the set of tips $\mathcal{T}$ as concisely (efficiently) and thoroughly as possible.

We first need to define when a review R covers a tip t. Reviews and tips are of different granularity. A tip is short and concise, usually making a single point, while a review is longer and multi-faceted, discussing various aspects of an entity. Intuitively, a review covers a tip, if the point made by the tip appears within the text of the review. To make this more precise, we break a review into sentences, which are semantic units with granularity similar to that of the tips. Given a sentence s and a tip t we define a binary matching function $\mathcal{F}$ such that $\mathcal{F}(s,t) = 1$ if s and t are sufficiently similar, and zero otherwise. Similarity between s and t means that s and t talk about the same thing, and we can think of one as covering the content of the other.

If a sentence s and a tip t are matched, then we say that s covers t. We will say that a review R covers a tip t if there is a sentence $s \in R$ that is matched to the tip t. We define the coverage of a review R as the number of tips covered by R. We also define the notion of the efficiency of a review R as the fraction of sentences in R that cover at least one tip. Our goal is to select a set of reviews that, collectively, have high coverage and high efficiency. Intuitively, this corresponds to a compact and comprehensive set of reviews covering most aspects of an item, while avoiding being verbose. In Section 3 we formulate the coverage problems considered in this paper, and present algorithms for constructing a solution.

The notion of similarity used in the definition of the function $\mathcal{F}$ is critical for the successful identification of true matches. We consider three different notions of similarity between a sentence s and a tip t: syntactic similarity, where we require s and t to share common vocabulary; semantic similarity, where we require s and t to share common concepts; sentiment similarity, where we require s and t to share common sentiment (positive or negative). We define a methodology for matching a sentence with a tip that takes into account all three of these different similarity definitions. Our methodology is described in detail in Section 4.

We evaluate experimentally the two parts of our approach, the mapping between reviews and tips and the output of the review selection process, using real data collected from Foursquare and Yelp. We evaluate the quality of the mapping, and we study the tradeoff between coverage and efficiency both quantitatively and qualitatively. The experimental analysis is presented in Section 5.

Contributions. Although the content of micro-blogging sites has been studied extensively, micro-reviews is a source of content that has been largely overlooked in the literature. In this paper we study micro-reviews, and we show how they can be used for the problem of review selection. To the best

of our knowledge we are the first to mine micro-reviews such as Foursquare tips and combine them with full-text reviews such as Yelp reviews. Our work introduces a novel formulation of review selection, where the goal is to maximize coverage while ensuring efficiency, leading to novel coverage problems. We propose heuristic algorithms for these problems, and study them experimentally, demonstrating quantitatively and qualitatively the benefits of our approach.

2. RELATED WORK

Our problem formulation, as far as we know, is novel, both in terms of the objective of covering micro-reviews, as well as in terms of the efficiency constraint. There are however related problem formulations that we discuss below.

Mining Reviews. Recently, there is a line of work that deals with the selection of a "good" set of reviews. In [10], the objective is to select a set of reviews that cover all attributes (for a given set of attributes). In [24], the objective is refined to also include both the positive and negative aspects of each attribute. The work in [9] further seeks to preserve the underlying distribution of positive and negative comments in the reviews. In [26], the objective is to cover more diversified opinion clusters, rather than just positive or negative. Related to review selection, [23] considers the problem of selecting a good set of photos based on quality, diversity, and coverage.

Our work is along the same lines, but is distinct in two ways. *First*, in terms of formulation, we seek to represent an underlying collection of micro-reviews, rather than attributes. *Second*, in terms of approach, while prior work relies on some variant of coverage formulation, ours is distinct in introducing the efficiency requirement. To compare against this class of approaches which focus on coverage but not efficiency, we will compare against a max coverage algorithm as a baseline in Section 5. There also exists a variant of max coverage called budgeted max coverage [6] where the constraint is a total cost that cannot be exceeded. Our coverage formulation is different in how both constraints of cost and count apply, and in how the total cost is computed.

Related to the notion of finding a "good" set of reviews is the problem of determining the quality of each individual review [13]. Sites such as Amazon or Yelp allow users to rate each review by its helpfulness or usefulness. Most review ranking works rely on a supervised regression or classification approach, using the helpfulness votes as the target class [3, 7, 12]. One possible formulation to produce a set of reviews is to first rank all the reviews based on individual merits, and then selecting the top K. The weakness of this formulation is that it ignores the potential similarities among the top reviews. It may well be that the top few reviews all represent the same information. For comparison, we introduce a baseline called *Useful* in Section 5, which ranks reviews by its usefulness votes, and selects the top K.

Our work is also related to the problem of review summarization, where the goal is to gain a quick overview of the underlying corpus of reviews. Existing approaches vary in the kind of summary they produce. In [4, 27], the summary is a list of features, the statistics of positive and negative opinions, as well as some example sentences. In [2, 18], the summary is a list of short phrases. If we treat a review as a document, the summary could also take the form of a subset of sentences from the underlying documents [11]. Different from these review summarization works, our objective is closer to micro-reviews summarization (using reviews).

Mining Micro-Reviews. Compared to the wealth of related work on reviews, there has not been as much interest in micro-reviews within the research community. One related work focuses on very short comments on eBay left by buyers about sellers [14], but the problem there was to extract aspects from the comments. There are also works [5, 8] on analyzing opinions in micro-blogging services such as Twitter. However, because Twitter is a general micro-blogging platform, these opinions are usually about more general concepts (e.g., brands, hashtags) rather than specific entities (e.g., products, restaurants). Unlike Foursquare tips, tweets are not attached to any entity, and it is difficult to separate "reviews" from other types of content in Twitter.

Most of the previous work on Foursquare or other mobile check-in services does not view them as a source of micro-reviews, but rather as location-based social networks (LBSN), and it addresses problems such as mining user profiles [25] and movement patterns [20], or protecting the privacy of the users' movement patterns [22].

3. REVIEW SELECTION

In this section we formulate the review selection problem. We will model this problem as a coverage problem where the goal is to select a small set of reviews that cover as many of the tips as possible with as few sentences as possible. That is, we want to maximize both the *coverage* and the *efficiency* of the selected set of reviews, by requiring that there is little content that is not related to at least one tip.

In order to define when a review covers a tip we assume that we are given a mapping between review sentences and tips. We view a review R as a set of sentences $R = \{s_1, ..., s_{|R|}\}$, and we use $\mathcal{U}_s$ to denote the union of all review sentences from the reviews in $\mathcal{R}$. The mapping is defined as a matching function $\mathcal{F} : \mathcal{U}_s \times \mathcal{T} \rightarrow \{0, 1\}$, where for a sentence $s \in \mathcal{U}_s$ and a tip $t \in \mathcal{T}$ we have:

$$\mathcal{F}(s,t) = \begin{cases} 1 & \text{if } s \text{ and } t \text{ are similar} \\ 0 & \text{otherwise} \end{cases}$$

The notion of similarity between a sentence s and a tip t, and the conditions for matching are formally defined in Section 4. Intuitively, similarity implies that s and t talk about the same concept, using similar language, and having a similar viewpoint (positive or negative).

3.1 Coverage and Efficiency

We first give the definitions of coverage and efficiency. Given the collection of reviews $\mathcal{R}$ and the collection of tips $\mathcal{T}$, and the matching function $\mathcal{F}$, we define for each review R the set of tips $\mathcal{T}_R$ that are *covered* by at least one sentence of review R. Formally:

$$\mathcal{T}_R = \{t \in \mathcal{T} : \exists s \in R, \mathcal{F}(s,t) = 1\}$$

We say that R *covers* the tips in $\mathcal{T}_R$. We define the coverage $\text{Cov}(R)$ of review R as the number $|\mathcal{T}_R|$ of tips covered by the review R. We also define the *efficiency* $\text{Eff}(R)$ of the review R as the fraction of sentences in R that cover at least one tip. Formally:

$$\text{Eff}(R) = \frac{|\{s \in R : \exists t \in \mathcal{T}_R, \mathcal{F}(s,t) = 1\}|}{|R|}$$

We can extend these definitions to the case of a *collection* of reviews. For a set of reviews $\mathcal{S} \subseteq \mathcal{R}$, we define the coverage of the set $\mathcal{S}$ as:

$$\text{Cov}(\mathcal{S}) = |\cup_{R \in \mathcal{S}} \mathcal{T}_R|$$

We also define the normalized coverage $\overline{\text{Cov}}(\mathcal{S})$ as the fraction of tips covered by the set $\mathcal{S}$, that is, $\overline{\text{Cov}}(\mathcal{S}) = \frac{\text{Cov}(\mathcal{S})}{|\mathcal{T}|}$. This normalized notion is useful for comparing between different datasets, where the size of reviews and tips may vary.

Extending the definition of efficiency to a collection of reviews is a little more involved. We need a way to aggregate the efficiency of the individual reviews. We propose three possible definitions.

- **Minimum Efficiency:** In this case, the efficiency of a set of reviews $\mathcal{S}$ is defined as the minimum efficiency of any review in the set. Formally:

$$\text{Eff}_{\min}(\mathcal{S}) = \min_{R \in \mathcal{S}} \text{Eff}(R)$$

- **Average Efficiency:** In this case, the efficiency of a set $\mathcal{S}$ is defined as the average efficiency of the reviews in the set. Formally:

$$\text{Eff}_{\text{avg}}(\mathcal{S}) = \frac{\sum_{R \in \mathcal{S}} \text{Eff}(R)}{|\mathcal{S}|}$$

- **Bag Efficiency:** In this case, we view a collection of reviews $\mathcal{S}$ as a single review $R_{\mathcal{S}}$ consisting of the union of the sentences of the reviews. We then define the efficiency of the collection as the efficiency of $R_{\mathcal{S}}$. Formally, we have $R_{\mathcal{S}} = \cup_{R \in \mathcal{S}} R$, and $\text{Eff}_{\text{bag}}(\mathcal{S}) = \text{Eff}(R_{\mathcal{S}})$.

$\text{Eff}_{\min}$ is useful for imposing a stringent condition on the efficiency of the reviews in the set $\mathcal{S}$. For instance, by requesting that the minimum efficiency is above some threshold, we gain a guarantee that all reviews in the set obey the threshold. The other two definitions Eff_{avg} and Eff_{bag} are more flexible, because they consider the set $\mathcal{S}$ as a whole. This allows us to select some reviews with high coverage but slightly lower efficiency, if we can balance this choice with other reviews with high efficiency in the set. Eff_{bag} is different from Eff_{avg} in that it effectively gives longer reviews a higher weight in computing the aggregate efficiency of a set.

3.2 Max Coverage with Efficiency Constraints

Maximizing both coverage and efficiency is a bi-criterion optimization problem, which has no single optimal solution. We transform it into a maximization problem by constraining the efficiency, and asking for a maximum coverage solution. Formally, our problem is defined as follows.

PROBLEM 1 (EFFMAXCOVERAGE). *Given a set of reviews $\mathcal{R}$, a set of tips $\mathcal{T}$, the matching function $\mathcal{F}$ between review sentences and tips, and parameters α and K, select a set $\mathcal{S}$ of K reviews such that the coverage* $\text{Cov}(\mathcal{S})$ *of the set is maximized, while the efficiency of the set is at least α, that is* $\text{Eff}(\mathcal{S}) \geq \alpha$.

It is easy to see that the EFFMAXCOVERAGE is NP-hard. The proof follows from the fact that in the special case that $\alpha = 0$, the EFFMAXCOVERAGE problem is the same as the MAXCOVERAGE problem, where our goal is to simply select K reviews that maximize the coverage. Therefore, the EFFMAXCOVERAGE problem is NP-hard, and we need to look for approximation, or heuristic algorithms.

Our problem definition differs depending on the choice of the efficiency function. In the case that we use the $\text{Eff}_{\min}$ efficiency function we can show a further equivalence with the MAXCOVERAGE problem. Requiring that $\text{Eff}_{\min}(\mathcal{S}) \geq \alpha$ implies that each of the selected reviews must have individual efficiency of at least α. Therefore, we can again show an equivalence with the MAXCOVERAGE problem, where the universe of available reviews is restricted to the subset of reviews that have efficiency at least α.

It is well known that due to the submodularity property of the coverage function, the greedy algorithm that always selects the review whose addition maximizes the coverage produces a solution with approximation ratio $(1 - \frac{1}{e})$, where e is the base of the natural logarithm [19]. That is, the coverage of the greedy algorithm is at least a $(1 - \frac{1}{e})$ fraction of the coverage of the optimal algorithm. Therefore, we obtain the following lemma.

LEMMA 1. *The greedy algorithm for the* EFFMAXCOVERAGE *problem with the* $\text{Eff}_{\min}$ *efficiency function has approximation ratio* $(1 - \frac{1}{e})$.

We could not determine an approximation bound for the other two variants of the efficiency function. In the following, we provide a heuristic algorithm which, as a special case, includes the greedy approximation algorithm for $\text{Eff}_{\min}$.

3.3 Algorithms

We present a general greedy algorithm for the EFFMAXCOVERAGE problem shown in Algorithm 1. The algorithm proceeds in iterations each time adding one review to the collection $\mathcal{S}$. At each iteration, for each review R we compute two quantities. The first is the *gain* $\text{gain}(R)$, which is the increase in coverage that we obtain by adding this review to the existing collection $\mathcal{S}$. The second quantity is the cost $\text{cost}(R)$ of the review R, which is proportional to the *inefficiency* $1 - \text{Eff}(R)$ of the review, that is, the fraction of sentences of R that are not matched to any tip. We select the review R^* that has the highest gain-to-cost ratio, and guarantees that the efficiency of the resulting collection is at least α, where α is a parameter provided in the input. The intuition is that reviews with high gain-to-cost ratio cover many additional tips, while introducing little irrelevant content, and they are desirable to be added to the collection.

The cost of the review is parameterized by a value $\beta \in [0, 1)$, provided as part of the input, which controls the effect of efficiency in our selection of the review R^*. More specifically, the cost of a review is defined as follows:

$$\text{cost}(R) = \beta(1 - \text{Eff}(R)) + (1 - \beta)$$

When $\beta = 0$, the review selection is not affected by the efficiency of the reviews, but only by the coverage. For β close to 1 the effect of the efficiency on the review selection is maximized. Values in between regulate the effect of efficiency in our selection. The higher the value of β, the higher the value of coverage that is needed for a low-efficiency review to be included in the set. For example, for β close to 1, a review R_1 with efficiency 0.5 needs to have at least 250% times more coverage to be picked over another review R_2 with efficiency 0.8. For $\beta = 0.5$, R_1 only needs 25% more additional coverage to be picked over R_2.

Algorithm 1 The *EffMaxCover* algorithm.

Input: Set of reviews $\mathcal{R}$ and tips $\mathcal{T}$; Efficiency function Eff; Integer budget value K, parameters α,β.
Output: A set of reviews $\mathcal{S} \subseteq \mathcal{R}$ of size K.

$\mathcal{S} = \emptyset$
while $|\mathcal{S}| < K$ **do**
  for all $R \in \mathcal{R}$ **do**
    $\text{gain}(R) = \text{Cov}(\mathcal{S} \cup R) - \text{Cov}(\mathcal{S})$
    $\text{cost}(R) = \beta(1 - \text{Eff}(R)) + (1 - \beta)$.
  end for
  $\mathcal{E} = \{R \in \mathcal{R} : \text{Eff}(\mathcal{S} \cup R) \geq \alpha\}$
  if $\mathcal{E} == \emptyset$ **then**
    break
  end if
  $R^* = \arg\max_{R \in \mathcal{E}} \text{gain}(R)/\text{cost}(R)$
  $\mathcal{S} = \mathcal{S} \cup R^*$
  $\mathcal{R} = \mathcal{R} \setminus R^*$
end while
return $\mathcal{S}$

We obtain different algorithms for different choices of the efficiency function. We study these different variations in detail in the experimental analysis. Note also that by varying the parameters α and β we can obtain some existing algorithms as special cases. For $\alpha = 0$ and $\beta = 0$ we obtain the greedy algorithm for the MaxCoverage problem. We refer to this algorithm as *MaxCover*. For $\beta = 0$ we obtain the greedy approximation algorithm for the case of the Eff_{min} efficiency function.

4. MATCHING REVIEWS AND TIPS

In this section we define the matching function $\mathcal{F}$ used in Section 3 for the definition of the coverage problem. We want to match a sentence s and a tip t if they convey a similar meaning, and therefore one can be seen as covering the content of the other. We now consider the criteria for making the matching decision. The first criterion, considers the sentence and the tip as collections of words. If they share a substantial subset of textual content then we assume that they convey a similar meaning. In this case we say that they have high *syntactic similarity*. The second criterion considers the concept that is discussed. A sentence and a tip may discuss the same concept (e.g., a menu dish), but use different words (e.g., soup vs. broth). In this case we say that they have high *semantic similarity*. Finally, reviews as well as tips, express the opinions of their respective authors. Hence, in addition to sharing similar keywords and concepts, we would also like a matching sentence-tip pair to share the same sentiment (positive or negative). In this case we say that they have high *sentiment similarity*.

In the following, we elaborate further on each of the above three types of similarity, and how they can be defined and measured. We then describe how to combine them into the matching function $\mathcal{F}$.

Syntactic similarity (SynSim). A review sentence and a tip are syntactically similar if they share important keywords. For example, a review sentence and a tip about the same Japanese restaurant both use distinctive words such as "ramen" and "noodle" when referring to a specific dish. A well-established model for keyword similarity is the vector space model [16]. Each review sentence s, and each tip t, are associated with vectors $\mathbf{s}$ and $\mathbf{t}$ respectively. The dimensionality of the vectors is the size of the vocabulary. Each vector entry signifies the importance of the corresponding word. The degree of similarity between the sentence and the tip is then measured as the cosine similarity [16]. Therefore we have:

$$\text{SynSim}(s, t) = \text{cosine}(\mathbf{s}, \mathbf{t}).$$

To compute the importance weights for the words we form a corpus of documents, where each document represents an entity (e.g., restaurant) and it consists of all the tips about this entity. We then use the standard *tf-idf* [16] weighting scheme for determining the importance of a word. The term frequency *tf* is the number of times the word appears in the entity document, while the inverse document frequency, *idf*, is determined by the number of different entity documents in which the word appears. The important words are those frequently used to describe an entity, and unique to the entity.

Semantic similarity (SemSim). A review sentence and a tip are semantically similar, when they are describing the same concept, even if they do not use exactly the same keywords. For instance, when discussing ramen noodles, some may choose to use "broth", while others use "soup", although both refer to the same concept. There are two main challenges in determining semantic similarity: first, identifying automatically concepts that are important to each entity; second, finding the words that are used to describe the concepts in text. To deal with these challenges, we seek an unsupervised approach that can work across different domains. Inspired by the work in text mining, we propose to discover the latent concepts from text using topic modeling.

While there are several potential topic models, here we describe an approach based on the well-known Latent Dirichlet Allocation (LDA) [1]. For illustration, in Table 1, we show an example of topics discovered from the Foursquare tips of a couple of restaurants in New York. Due to space limitation, we show five out of 20 topics learned from each restaurant's tips. Ippudo[2] is a Japanese restaurant serving ramen and pork buns. Some of the topics describe menu dishes (101), waiting time (102), drinks (104), and service (105). These are pertinent concepts in the restaurant domain. Similarly, for the fast-food joint Shake Shack[3], the topics include menu dishes (201), queue (202), dessert (203), and location (205). This small example serves to demonstrate that the topics do reflect the pertinent concepts in each restaurant.

Restaurant	Topic #	Top 5 keywords
Ippudo	101	ramen, pork, bun, modern, akamaru
	102	wait, time, hour, worth, ramen
	103	noodl, ramen, extra, order, flavor
	104	lyche, chill, citi, martini, tip
	105	great, servic, host, hair, curli
Shake Shack	201	burger, shack, shake, fri, chees
	202	line, wait, burger, worth, it', long
	203	custard, frozen, flavor, awesom, eat
	204	burger, spot, foodspot, shroom, shack
	205	park, madison, squar, stand, locat

Table 1: Example of topics for several restaurants

LDA associates each tip t with a probability distribution θ_t over the topics, which captures which topics are most im-

[2] https://foursquare.com/v/ippudo/4a5403b8f964a520f3b21fe3
[3] https://foursquare.com/v/shake-shack/40e74880f964a520150a1fe3

portant for a tip. Given the topics, and the corresponding language model for each topic as it is learnt from the tips, we can estimate the topic distribution θ_s for each review sentence s, which captures how well a sentence s reflects the topics being discussed in the corpus of tips. To measure the semantic similarity between a review sentence and a tip, we measure the similarity of the topic distributions θ_s and θ_t. A commonly used distance measure between two probability distributions is the Jensen-Shannon Divergence (JSD) [16]. Intuitively, a sentence and a tip are semantically similar if their topic distributions can describe each other well. Therefore, we have:

$$\text{SemSim}(s,t) = 1 - \text{JSD}(\theta_s, \theta_t).$$

Sentiment similarity (SentSim). A matching pair of review sentence and tip should also represent the same sentiment. Sentiment extraction from text is an active area of research [21]. Here, we cast the problem as a classification problem, where the goal is to predict the sentiment (positive or negative) of a sentence or a tip. We thus have two classes c^+ and c^-. We use a maximum entropy classifier (MEM) [15], which has been demonstrated to work well for sentiment classification in text [21], using N-gram features. Given a document d (a sentence or a tip), the MEM classifier outputs conditional probabilities $P(c^+|d)$ and $P(c^-|d)$ for the positive and negative classes, where $P(c^+|d) + P(c^-|d) = 1$.

Given the classifier output for a document d, we transform the probability $P(c^+|d) \in [0,1]$ into $\text{polarity}(d) = 2P(c^+|d) - 1$, in the range of -1 (extremely negative) to 1 (extremely positive). For $P(c^+|d)$ close to 1/2, the polarity is close to zero, which agrees with our intuition that in these cases the document has neutral polarity. We define the sentiment similarity between a sentence s and a tip t as the product of their polarities: it approaches 1 when the sentence and the tip's polarities are similar; it approaches -1 when their polarities are opposite; it approaches 0 when the tip or the sentence is neutral. Therefore, we have:

$$\text{SentSim}(s,t) = \text{polarity}(s) \times \text{polarity}(t)$$

Matching Function. Having defined the three main criteria for matching (syntactic, semantic, and sentiment), we would like to combine them to determine whether a review sentence s and a tip t match or not. One principled way to combine the three criteria is through a supervised binary classification framework, with two classes *match* and *non-match*, based on the three features we defined above: syntactic similarity $\text{SynSim}(s,t)$, semantic similarity $\text{SemSim}(s,t)$, and sentiment similarity $\text{SentSim}(s,t)$. For a sentence-tip pair (s,t) the classifier estimates the matching probability $P(s,t)$. The binary mapping function $\mathcal{F}(s,t)$ is thus defined in terms of the matching probability, using on a threshold η, as follows:

$$\mathcal{F}(s,t) = \begin{cases} 1 & \text{if } P(s,t) > \eta \\ 0 & \text{otherwise} \end{cases}$$

We discuss the choice of the threshold η in the experiments.

5. EXPERIMENTS

The objective of the experiments is to showcase the effectiveness of the proposed approach in finding a set of reviews that cover as many tips as possible, in an efficient manner. First, we will describe the real-life dataset used in the experiment. This is followed by an evaluation of the matching process described in Section 4. We then investigate how the coverage algorithms proposed in Section 3 behave under different parameter settings, as well as how they compare against the baselines. Our focus here is on effectiveness, rather than speed, as the matching can be done offline, and the greedy algorithm for review selection is fast.

5.1 Dataset

The experiments require data coming from two different sources (reviews and micro-reviews), but concerning the same set of entities. We pick the domain of restaurants, because it is one of the few domains where there are already active platforms for reviews as well as for micro-reviews. For reviews, we crawl Yelp.com to obtain the reviews of the top 110 restaurants in New York City with the highest number of reviews as of March 2012. For micro-reviews, we crawl the popular check-in site Foursquare.com to obtain the tips of the same 110 restaurants. However, some of the restaurants in Foursquare.com have too few tips, which may not adequately reflect the restaurant's information. Therefore, we retain only the 102 restaurants with at least 50 tips each. For these 102 restaurants, we have a total of 96,612 reviews, with a minimum of 584, and a maximum of 3460 per restaurant. We also have a total of 14,740 tips, with a minimum of 51, and a maximum of 498 per restaurant. Note that we get the *full* set of reviews and tips of each restaurant at the time of extraction, and that these are the realistic sizes of the real-world data. It is also important to note that every restaurant is a distinct instance of the coverage problem.

5.2 Matching

Matching between a review sentence and a tip is by itself a very challenging problem. Our objective in this experiment is to establish that we achieve a reasonable level of quality in matching, such that the reviews selected by the coverage algorithms would be a good reflection of the covered tips.

To build the matching classifier, we generate the three real-valued features described in Section 4. For semantic similarity, we train LDA [1] topic models using the MALLET toolbox [17]. Because topic modeling is probabilistic, we average the semantic similarity over ten runs. To determine the sentiment polarity of each sentence and tip, we train a sentiment classifier using the Stanford Classifier toolkit [15] with textual features (word and letter n-grams).

To train the matching classifier, we sample 20 entities, and for each entity we sample 50 sentence-tip pairs sharing at least one common word. We assume no match otherwise. For these 1,000 pairs, we get three judges to label whether the pairs match in meaning, and take the majority label as the ground truth. Finally, we use the real-valued features and the majority labels to train the matching classifier using the MEM classifier from [15]. Based on the feature weights learned by the classifier, we find that among the three features, semantic similarity is the most important, followed by syntactic, and lastly sentiment.

To validate the effectiveness of the matching classifier, we conduct a five-fold validation, with 80:20 split between training and testing in each fold. As metrics, we use precision and recall at the pair level. Precision is the fraction of true matching pairs within the set of classified matching pairs. Recall is the fraction of true matching pairs found by the

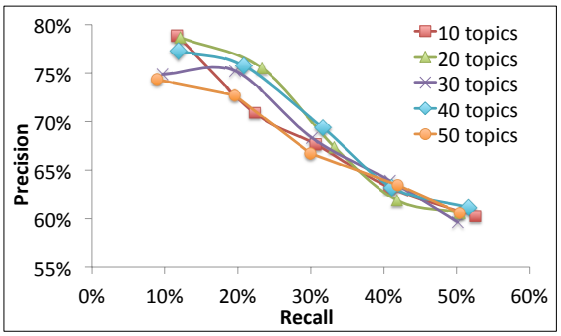


Figure 1: Matching: Precision-Recall Curve

classifier within the set of all true matching pairs. Because the objective of matching is to determine which review sentence will match a tip well, it is important to gain a high precision, so we can be confident that the reviews discovered by the coverage algorithms will reflect the underlying tips.

Number of topics. We study the performance of matching classifier as we vary the number of topics used for the semantic similarity. In Figure 1, we plot the precision-recall curve for $\eta = 0.65$ (discussed below). Besides showing the regular trade-off between precision and recall, it also shows that the effect of the number of topics is not significant. The performance for 20–40 topics is better than 10 (which may underfit), or 50 (which may overfit). The results for 20 topics are slightly better than the rest, especially in achieving higher precision, which is our main concern in the matching. In subsequent discussions, we show the results for 20 topics.

Threshold η. We also experiment with different values for the threshold η on the probability of matching $P(s,t)$. Table 2 shows the precision and recall of the matching classifier at different values of η. If we were to skip the matching classification, and simply take all the pairs with at least one common word as matching, we get a precision of only 43%, which means more than half of all matching pairs would be incorrect. As we increase the threshold η, the precision improves significantly. If we would like at least three-quarters of matching pairs to be correct, we need to put the threshold at 0.65 of higher. At this threshold, the recall is relatively low at 23%, but this can be compensated by the fact that a tip may be covered by many different sentences.

The last column shows the percentage of tips that can be covered by at least one sentence. At 0.65, we cover 83.5% of all tips, a substantial subset. For subsequent experiments on coverage, we will present results for $\eta = 0.65$.

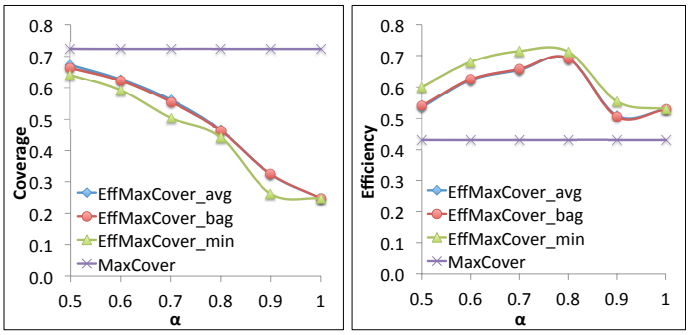
Threshold η	Matching Pairs		Coverable Tips
	Precision	Recall	
0.70	78.6%	12.1%	72.3%
0.65	75.5%	23.3%	83.5%
0.60	67.4%	33.2%	89.7%
0.55	61.9%	41.8%	93.4%
0.50	60.6%	50.4%	95.9%
All	42.9%	100.0%	100.0%

Table 2: Matching Classifier

To get an intuitive sense of the matching quality, we show some examples of matching pairs for the restaurant Ippudo in Table 3. The first pair discuss the pork buns, which is a specialty of the restaurant. The other pairs both discuss the waiting time, but while the second pair sound positive, the third pair show negative sentiment. These examples showcase how the features, i.e., syntactic, semantic, and sentiment similarity, help to identify relevant matching pairs.

ID	Review Sentence - Tip Matching Pair		$P(s,t)$
1	Review	The best part was the pork bun that was delicious.	0.839
	Tip	pork buns are so tasty!	
2	Review	I went with a group of 5 on a Saturday for lunch, and the wait was only 15 minutes.	0.792
	Tip	go for lunch - little to no wait!	
3	Review	the ramen was good, but minus one star for priciness and another -1 star solely based on the fact that i've had better ramen for half the price.. and the fact that i had to wait like 2 hours.	0.650
	Tip	Not worth the wait. $15 ramen with $2 toppings are a turn-off. You can walk a few blocks to Rai Rai Ken and get better ramen.	

Table 3: Example of matching pairs for Ippudo, NY



a. Coverage b. Efficiency

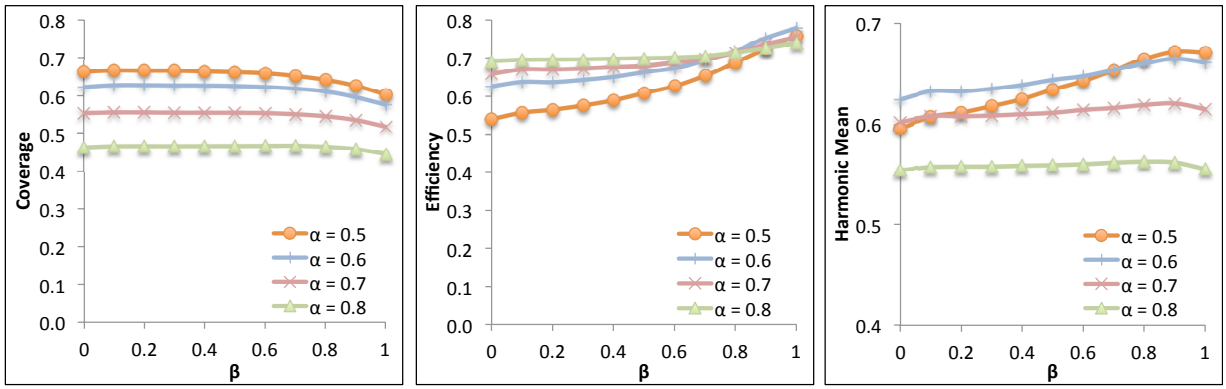
Figure 2: Varying α for $\beta = 0$

5.3 Coverage & Efficiency

The objective of these experiments is to showcase the efficacy of the proposed *EffMaxCover* algorithm at finding the top K reviews with high coverage of tips, while satisfying the efficiency constraint. We will first show results for $K = 5$, before we investigate the effect of varying K.

The input to the algorithms is the sentence-tip matchings generated for all 102 restaurants in the dataset. To avoid degenerate cases of reviews that achieve very high efficiency simply by being very short, we restrict ourselves to reviews of at least five sentences. Our evaluation is based on the normalized coverage, defined as the fraction of tips that are covered by the top K reviews over the total number of coverable tips, and the average efficiency $\text{Eff}_{\text{avg}}(\mathcal{S})$, defined as the average efficiency of the individual reviews in the top K. To represent the results for all the restaurants, we average the coverage and efficiency values accross restaurants.

Baseline: MaxCover. We first establish the baseline level of performance by *MaxCover*, which also has the objective of maximizing coverage, but does not consider the efficiency constraint. Because *MaxCover* is not constrained in the review selection, it obtains a relatively high coverage of 0.72, which is also the ceiling for *EffMaxCover* (because of the efficiency constraint). *MaxCover*'s efficiency is only 0.43, and this is the floor for *EffMaxCover* that searches for a more efficient set of reviews.



a. Coverage b. Efficiency c. Harmonic Mean

Figure 3: Varying β for $\alpha \in [0.5, 0.8]$ for *EffMaxCover*$_{bag}$

EffMaxCover: Varying α. There are two ways in which *EffMaxCover* controls the efficiency of the selected set of reviews. The first is by the threshold α, which guarantees the efficiency of the set is at least α. The second is by the parameter β which controls the sensitivity of the selection process to the efficiency of the next review to be added to the set. To isolate the effect of α, we first fix $\beta = 0$, making the cost a constant, independent of the efficiency. Since *MaxCover* already has an efficiency of 0.43 and the "optimal" coverage, we will focus on $\alpha > 0.43$, and investigate whether *EffMaxCover* can achieve a higher efficiency without much reduction in coverage. We vary α from 0.5 to 1.0, and plot the coverage and efficiency in Figure 2.

Figure 2(a) shows that as α increases, the coverage of *EffMaxCover* algorithms decrease. Due to the constraint on the aggregate efficiency being at least α, we miss out on higher-coverage, but lower-efficiency reviews. Figure 2(b) shows that efficiency at first increases with α, because the reviews selected tend to be of increasingly higher efficiency. At some point though, when $\alpha > 0.8$, the efficiency decreases, because this requirement becomes too stringent, and many restaurants do not have reviews that meet this requirement.

Among the different ways of aggregating efficiency for *EffMaxCover*, we observe *EffMaxCover*$_{avg}$ and *EffMaxCover*$_{bag}$ perform very similarly. On the other hand, *EffMaxCover*$_{min}$ performs differently. It tends to have higher efficiency but lower coverage, because every review selected has to meet the efficiency threshold, reducing the set of candidate reviews available, whereas the other two algorithms consider the efficiency of the whole set and may pick some reviews with high coverage, but with efficiency slightly below α if the reviews already in the set have high efficiency.

EffMaxCover: Varying β. Having fixed parameter α, we now study the effect of parameter β on the performance of the algorithm. Figure 3 shows how the coverage and efficiency change as β increases from 0 to 1 for *EffMaxCover*$_{bag}$ (the curves for other variants are similar and not shown due to space limitation). Following the previous discussion, we plot the curves for the values of α between 0.5 to 0.8.

At $\beta = 0$, the cost is a constant, and we rely entirely on α to maintain efficiency. As we increase β, the greedy selection of reviews will increasingly be sensitive to the cost (loss in efficiency). Figure 3 shows that for all values of α, as β increases, the efficiency increases while the coverage decreases. Interestingly, the gain in efficiency outpaces the loss in coverage. For example, for $\alpha = 0.5$, from $\beta = 0$ to $\beta = 1$, efficiency increases from 0.54 to 0.76 (efficiency gain of 0.22), while the coverage reduces from 0.66 to 0.60 (coverage loss of 0.06). This shows that β is an effective way to gain efficiency with minimal loss in coverage.

In order to have a single metric that balances the coverage vs. efficiency trade-off, inspired by the F1 measure in information retrieval, we use the harmonic mean of the two:

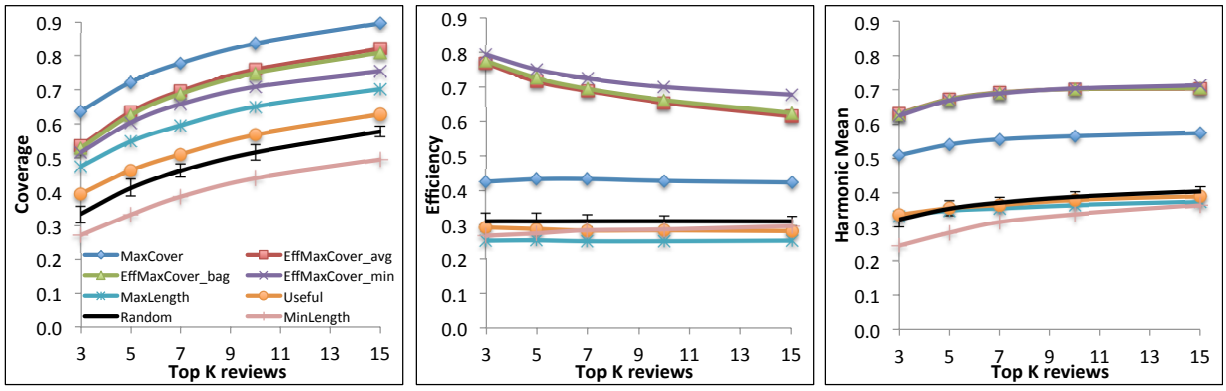
$$\text{HMean}(\mathcal{S}) = \frac{2 \times \overline{\text{Cov}}(\mathcal{S}) \times \text{Eff}_{\text{avg}}(\mathcal{S})}{\overline{\text{Cov}}(\mathcal{S}) + \text{Eff}_{\text{avg}}(\mathcal{S})}$$

Figure 3(c) plots the harmonic mean when varying β and α values. It shows that $\alpha = 0.5$ and $\alpha = 0.6$ tend to have a better balance between having high coverage and high efficiency. Of all the points in Figure 3(c), the combination with the highest harmonic mean of 0.67 is $\alpha = 0.5$ and $\beta = 0.9$, which yields a coverage of 0.63 and an efficiency of 0.72. Subsequently, we will use this setting for *EffMaxCover*.

EffMaxCover: Varying K. We now compare the performance of *EffMaxCover* to *MaxCover* as well as to other baselines, for varying top $K \in [3, 15]$ reviews. We consider the following additional baselines. *MaxLength* selects the longest K reviews, with the intuition that longer reviews may cover more tips. Conversely *MinLength* selects the shortest K reviews (not less than five sentences), with the intuition that shorter reviews may be more efficient. Yelp reviews may also be voted by users as being useful, and we consider the K reviews with the highest number of usefulness votes as another baseline *Useful*. Finally, to emphasize the statistical significance of the results, we also compare to the performance of *Random*, which selects K reviews randomly. For *Random*, we average the coverage and efficiency across 1,000 random runs, and plot the median, as well as the error bar (min and max).

Figure 4(a) shows how coverage varies with K for various methods. As expected, *MaxCover* has the highest coverage, followed closely by the *EffMaxCover* variants. *MaxLength* and *Useful* also do better than *Random*, but worse than *EffMaxCover*. *MinLength* has the lowest coverage, as it has very few sentences to capture the tips.

Figure 4(b) shows that the efficiency of *EffMaxCover* algorithms is by far superior to all the baselines. This underlines the effectiveness of *EffMaxCover* in finding efficient reviews. The efficiency tends to decrease slightly with increasing K, which is expected as it gets increasingly more difficult to find high-coverage and high-efficiency reviews after each selection. Interestingly, the efficiency of *MaxLength*



a. Coverage b. Efficiency c. Harmonic Mean

Figure 4: Varying K for $\alpha = 0.5$, $\beta = 0.9$

and *Useful* fall below that of *Random*, which could be due to the length of the reviews, resulting in having many sentences that may not represent any tip. *MinLength* is more efficient than *MaxLength*, but is also worse than *Random*. This suggests that being short alone is not sufficient if it does not also capture the tips well.

To emphasize the efficacy of *EffMaxCover* at achieving both coverage and efficiency, we plot the harmonic mean of coverage and efficiency in Figure 4(c). It shows how the three *EffMaxCover* variants outperform the rest significantly, followed by *MaxCover*. *MaxLength* and *Useful* are no better than *Random*, whereas *MinLength* is the worst.

Qualitative Analysis. In addition to quantitative study, we also conduct a qualitative analysis involving three human judges who are not related to this paper. To each judge, we show the top 3 reviews selected by an algorithm for a sample of 20 restaurants, and ask the judge to choose which aspects are mentioned in the reviews from a manually hand-picked list of aspects. Because the objective of this analysis is to investigate the trade-off between coverage and efficiency, we focus the comparison on two methods: the $EffMaxCover_{bag}$ algorithm as a representative of the *EffMaxCover* variants, and the *MaxCover*, as the closest competitor.

Table 4 shows that on average, the judges identify 5.1 aspects for *MaxCover*, and 3.6 aspects for $EffMaxCover_{bag}$. This lower coverage of aspects is expected, and consistent with the previous experiments. On the other hand, the reviews selected by $EffMaxCover_{bag}$ are much more compact, with an average of 24.7 sentences total in three reviews, as compared to the lengthy 121 sentences by *MaxCover*. This suggests a gain in efficiency. If we look at the density of information covered, and determine the ratio of aspects covered per sentence, the third column of Table 4 shows that $EffMaxCover_{bag}$ has much higher density of 0.15 aspects per sentence, as compared to 0.04 by *MaxCover*.

Algorithm	*Aspects*	*Sentences*	*Aspects per sentence*
EffMaxCover_{bag}	3.6	24.7	**0.15**
MaxCover	5.1	121.0	0.04

Table 4: User Study

5.4 Case Study

To illustrate the different types of reviews selected by the various criteria, as a case study, we show an example of the top review selected by each algorithm for the venue *53rd and 6th Halal Cart*. This is a food cart serving middle-eastern fare in New York, well-known for its meat dishes and sauces. In Figure 5, we show the top review selected by *EffMaxCover* (all three variants selected the same), *MaxCover*, *Useful*, and *MinLength*. Due to space limitation, we cannot reproduce *MaxLength* here, but we refer the reader to the following link: http://www.yelp.com/biz/53rd-and-6th-halal-cart-new-york#hrid:s1opbJu3mS3L-DSsOXmIYQ.

Figure 5(a) shows that *EffMaxCover* selects a compact review, which describes the main attributes of the place: a food cart, popular chicken lamb combo, sauces, and long lines. Figure 5(b) shows that *MaxCover*'s top review also covers these attributes, but with a very long review. Parts of the review are not to the point. For instance, the first quarter ("background") does not concern the restaurant directly. Figure 5(c) shows that *Useful*'s top review also covers these attributes, but not as compactly as *EffMaxCover*, with side references to Paris Hilton and Victoria Secret that are not pertinent to the restaurant. A similar conclusion can be drawn for *MaxLength* as well. *MinLength*'s top review (Figure 5(d)) is very short and only covers the generics ("fast", "cheap", "good"), without getting into helpful details such as the dishes and the sauces, like the other reviews above.

6. CONCLUSION

In this paper, we introduce the use of micro-reviews for finding an informative and efficient set of reviews. This selection paradigm is novel both in the objective of micro-review coverage, as well as in the efficiency constraint. The selection problem is shown to be NP-hard, and we design a heuristic algorithm *EffMaxCover*, which lends itself to several definitions of aggregate efficiency. The results are evaluated over a corpora of restaurants' reviews and micro-reviews. Experiments show that *EffMaxCover* discovers review sets consisting of reviews that are compact, yet informative. Such reviews are highly valuable, as they lend themselves to quick viewing over mobile devices, which are increasingly the predominant way to consume Web content.

Acknowledgments

The work is supported by the National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office. This work has been supported by the Marie Curie Reintegration Grant project titled JMUGCS which has received research funding from the European Union.

The best late night street food. Get the combo plate. It's lamb and chicken grilled and served with rice and lettuce. You can cover it in the white sauce (what really makes it delicious) and some hot sauce on the side of the cart. Careful, the hot sauce is pretty hot. The line can be long at times and it's always at the same single cart, although there are other similar carts around. You definitely get the freshest food at this cart since they're constantly turning out food while it might be sitting a little while at the other carts.

(a) EffMaxCover

The little halal food cart that could. Feed New York's hordes of drunken masses, that is... BACKGROUND ——————————- Back when I kinda sorta moved here in January, I had no idea this particular cart even existed. A few weeks later at work, a colleague of mine posed a trivia question for a prize: "What is New York's most famous landmark?" A) Empire State Building B) Statue of Liberty C) Central Park D) Brooklyn Bridge or E) Halal Food Cart at 53rd & 6th? "Huh?" I pondered aloud, "What kind of choice is that? There are halal food carts all over Manhattan." "You don't understand. This cart is special. It's far above all the other carts. It's all about the chicken and rice," was the enthusiastic response. "People wait in the longest line for it. And there are the sauces! You must have the sauce. You don't understand. It's the best thing ever!" "Uh, you're quite excited about this chicken and rice. And sauce. So, when do you usually eat at this cart?" I asked. "Late at night after hitting the bars and clubs," was the immediate response. "I see." And so I remained skeptical, for anything tastes amazing at 3 in the morning and several drinks later... FAST FORWARD TO JUNE... ——————————————————- I have the day off. For some reason I am also watching the Today Show. And for some reason, Jenna Wolfe is waiting in line at the halal food cart at 53rd and 6th at lunchtime, boring the crap out of anyone within earshot, to make a point that the line for this cart is insanely long, even in the middle of the day with a good majority of the patrons in suits. "Why are you waiting 40 minutes for food from a cart?" she asked someone. "Cause it's amazing?" was the response. Still, I remained skeptical... FAST FORWARD TO JULY 4th ——————————————————— Heat! Humidity! Sitting on 12th Ave for hours on end! Fireworks! So those all happened. I was with friends from out of town. They were tired and wanted to go back to where they were staying. For a while I thought I would head back to Jersey myself. But then... "You're already in midtown...." my brain said. "Now is your chance to check this cart for yourself." My autopilot took over. I bid my friends adieu at Times Square and made a beeline up 6th, passing countless other halal food carts along the way. "This is stupid," I thought to myself. Yet I kept walking. And I reached the corner of 53rd & 6th. And there were 50 people in line. And I got in the back of it. "Still stupid," I thought. Yet I kept waiting. And waiting. And waiting. Eventually I got the front. And I noticed the yellow-shirted guys were working at a furious pace. Slicing shawrma off the spit. Scooping up chopped chicken. Cutting up pita. But there was no menu to be found. What the hell do I get? Uh... "Chicken & lamb!" I blurted out. The guy quickly nodded, turned around, and yelled it out. And in no time flat a round foil container full of rice was covered in chicken, lamb, and pita slices. And then, it was in my hands. I looked at it. "This....is it?!" It didn't look like it was something half of New York would be tripping themselves over, but what do I know? I then looked to the right of the cart, and the person who ordered before me was squeezing the life out of a plastic bottle onto his bowl, completey drowning it in white sauce. "The sauce," I remembered. "It's all about the sauce." And so I walked over, squirted a fair amount of white sauce. And a more than hefty amount of the spicy red sauce, too. And then I sat down. On the sidewalk. And proceeded to eat... AND....? ————- HOLY MOTHER OF HALAL FOOD CARTS THAT RED SAUCE IS HOT! It was fiery and amazing. A hellacious sweet symphony in my mouth. I then started to hiccup uncontrollably. And also started sweating profusely. But I couldn't stop eating. Oh my God. I ate. And I ate. No one was around to tell me to slow down. Whatever. Mixed the red sauce with the white sauce with the rice and chicken and everything else. And kept eating. Was the food that good? Or was I that hungry? (I did walk 8 miles earlier that day). Or was there something else in it? Crack? I couldn't tell. But I plowed through the whole container in one sitting. Stone cold sober. I looked around me. Countless other people of every color and creed were doing the same thing. This halal food cart, on the 4th of July, truly has brought everyone together. I would have felt more sentimental and patriotic, but it was late. SO WHY NOT 5 STARS? ———————————————————- I think I made the mistake of not eating at any other halal food cart before this. I'll need to do so, for comparison's sake, and adjust my ratings accordingly. TIP —— A second cart is at 52nd & 6th. Same yellow-shirted guys, probably to help with the overflow.

(b) MaxCover

One thing I admire about New York is how the city embraces street food as integral to its culinary landscape as its abundant highly decorated Michelin-rated restaurants. And among the gajillion carts spread across almost every corner, the legendary Halal cart on 53rd and 6th appears to be the undisputed people's champion – armed with its own website and over 2,000 yelp reviews with a solid 4.5 rating. You're probably more likely to see Paris Hilton win an Oscar than see this popular cart without a line that spans at least half the block. Their slogan is "We are different." Indeed. The line was so long, you would have thought Victoria Secret models were giving free blow...dryers. Thanks to yelp advice, I steered clear of impostors from several other corners wearing the same color shirts and made my way to the SE Corner of 53rd and 6th. Yes, the food was worth the wait (and the wait wasn't even that long). Cheap and good. The lamb plate was pretty fantastic. Though my yellow rice was a bit of a grease fest (it's still cart food after all), it was still very good. The lamb was nice and tender as well, but the star of the show was that delicious, highly palatable white sauce. Mix it up with the huge pile of yellow rice, meat and warm pita ready and, boom...GOT HEEM! If you're into hot, the hot sauce was hotter than hades' armpit. Overall, best $6 I've spent in New York. Don't worry about the oft-repeated "tourist trap" moniker. Most locals I've met rave about this cart, including my colleagues who took me here for lunch, satisfaction guaranteed. Anyway, for $6, you can afford to take that chance.

(c) Useful

fast. cheap. and really good. what more can you ask for? definitely better than the halal cart on 50th and 6th. :)

(d) MinLength

Figure 5: Top review for *53rd and 6th Halal Cart*

7. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3, 2003.

[2] K. Ganesan, C. Zhai, and E. Viegas. Micropinion generation: an unsupervised approach to generating ultra-concise summaries of opinions. In *WWW*, 2012.

[3] A. Ghose and P. G. Ipeirotis. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *ICEC*, 2007.

[4] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, 2004.

[5] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *JASIST*, 60(11), 2009.

[6] S. Khuller, A. Moss, and J. S. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[7] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti. Automatically assessing review helpfulness. In *EMNLP*, 2006.

[8] E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis: The good the bad and the omg. In *ICWSM*, 2011.

[9] T. Lappas, M. Crovella, and E. Terzi. Selecting a characteristic set of reviews. In *KDD*, 2012.

[10] T. Lappas and D. Gunopulos. Efficient confident search in large review corpora. In *ECML/PKDD*, 2010.

[11] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *HLT*, 2010.

[12] Y. Liu, X. Huang, A. An, and X. Yu. Modeling and predicting the helpfulness of online reviews. In *ICDM*, 2008.

[13] Y. Lu, P. Tsaparas, A. Ntoulas, and L. Polanyi. Exploiting social context for review quality prediction. In *WWW*, 2010.

[14] Y. Lu, C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In *WWW*, 2009.

[15] C. Manning and D. Klein. Optimization, maxent models, and conditional estimation without magic. In *Tutorial at HLT-NAACL 2003 and ACL 2003*, 2003.

[16] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[17] A. K. McCallum. Mallet: A machine learning for language toolkit. In *http://mallet.cs.umass.edu*, 2002.

[18] X. Meng and H. Wang. Mining user reviews: from specification to summarization. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 2009.

[19] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-I. *Mathematical Programming*, 14(1):265–294, 1978.

[20] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in foursquare. In *ICWSM*, 2011.

[21] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP*, 2002.

[22] T. Pontes, M. Vasconcelos, J. Almeida, P. Kumaraguru, and V. Almeida. We know where you live: privacy characterization of foursquare behavior. In *UbiComp*, 2012.

[23] P. Sinha, S. Mehrotra, and R. Jain. Summarization of personal photologs using multidimensional content and context. In *ICMR*, 2011.

[24] P. Tsaparas, A. Ntoulas, and E. Terzi. Selecting a comprehensive set of reviews. In *KDD*, 2011.

[25] M. A. Vasconcelos, S. Ricci, J. Almeida, F. Benevenuto, and V. Almeida. Tips, dones and todos: uncovering user profiles in foursquare. In *WSDM*, 2012.

[26] W. Yu, R. Zhang, X. He, and C. Sha. Selecting a diversified set of reviews. In *APWeb*, 2013.

[27] L. Zhuang, F. Jing, and X.-Y. Zhu. Movie review mining and summarization. In *CIKM*, 2006.