Unsupervised Ontology- and Sentiment-Aware Review Summarization

Nhat X. T. Le(✉), Neal Young, and Vagelis Hristidis

Department of Computer Science and Engineering, University of California,
900 University Avenue, Riverside, CA 92521, USA
{nle020,neal.young}@ucr.edu,vagelis@cs.ucr.edu

Abstract. In this Web 2.0 era, there is an ever increasing number of customer reviews, which must be summarized to help consumers effortlessly make informed decisions. Previous work on reviews summarization has simplified the problem by assuming that aspects (e.g., "display") are independent of each other and that the opinion for each aspect in a review is Boolean: positive or negative. However, in reality aspects may be interrelated – e.g., "display" and "display color" – and the sentiment takes values in a continuous range – e.g., somewhat vs very positive. We present a novel, unsupervised review summarization framework that advances the state-of-the-art by leveraging a domain hierarchy of concepts to handle the semantic overlap among the aspects, and by accounting for different sentiment levels. We show that the problem is NP-hard and present bounded approximate algorithms to compute the most representative set of sentences or reviews, based on a principled opinion coverage framework. We experimentally evaluate the proposed algorithms on real datasets in terms of their efficiency and effectiveness compared to the optimal algorithms. We also show that our methods generate summaries of superior quality than several baselines in short execution times.

Keywords: Review summarization · Unsupervised extractive summarization · Online customer review · Aspect based sentiment analysis

1 Introduction

Online users are increasingly relying on user reviews to make decisions on shopping (e.g., Amazon, Newegg), seeking doctors (e.g., Vitals.com, zocdoc.com) and many others. However, as the number of reviews per item grows, especially for popular products, it is infeasible for customers to read all of them, and discern the useful information from them. Therefore, many methods have been proposed to summarize customer opinions from the reviews [5,9,13,17]. They generally either choose important text segments [13], or extract product concepts (also referred as aspects or attributes in other works), such as "display" of a phone, and customer's opinion (positive or negative) and aggregate them [5,9,17].

R. Cheng et al. (Eds.): WISE 2019, LNCS 11881, pp. 747–762, 2019.
https://doi.org/10.1007/978-3-030-34223-4_47

However, neither of these approaches takes into account the relationship among product's concepts. For example, assuming that we need the opinion summary of a smartphone, showing that the opinions for both *display* and *display color* are very positive is redundant, especially given that we would have to hide other concepts' opinion (e.g., "battery"), given the limited summary size. What makes the problem more challenging is that the opinion of a user for a concept is not Boolean (positive or negative) but can take values from a linear scale, e.g., "very positive", "positive", "somewhat positive", "neutral", and so on. Hence, if "display" has a positive opinion, but "display color" has neutral, the one does not subsume the other, and both should be part of the summary. Further, a more general concept may cover a more specific but not vice versa.

Our key contribution is a novel review summarization framework that accounts for the relationships among the concepts (product aspects), while at the same time supporting various sentiment levels. Specifically, we model our problem as a pairs coverage problem, where each pair consists of a concept and a sentiment value, and coverage is jointly defined on both of them. We show that the problem of selecting the best concepts and opinions to display is NP-hard even when the relationships among the concepts are represented by a Directed-Acyclic-Graph (DAG). For that, we propose bounded approximation algorithms inspired by well-studied graph coverage algorithms.

To summarize, the review summarization framework consists of the following tasks: *(a) Concept Extraction*: we build upon existing work for extracting hierarchical concepts (aspects) from reviews. *(b) Sentiment Estimation*: estimate the sentiment of each mentioned concept on a linear scale. *(c) Select k representatives*: depending on the problem variant, a representative is a concept-sentiment pair (e.g., "display" = 0.3), or a sentence from a review (e.g., "this phone has pretty sharp display") or a whole review. Our proposed selection algorithms can be used to select representatives at any of these granularities. Note that our summarization approach is unsupervised, thus does not require any labeled dataset which is expensive to create in a new domain.

Our contributions can be summarized as below:

- We propose a fresh perspective for the review summarization problem that exploits available concept hierarchies and a novel opinion coverage definition. We model the problem as a coverage optimization problem (Sect. 2) and show how to map a set of reviews to our model (Sect. 5.1).
- We prove that the problem is NP-hard and propose several efficient approximation algorithms with guaranteed bounds (Sect. 4).
- We carry out a thorough evaluation on the cost and time of our proposed algorithms. We experimentally evaluate our methods on real collections of online doctor patient reviews, using popular medical concept hierarchies [10], and corresponding concept medical extraction tools [1].
- We perform qualitative experiments on both online doctor patient reviews and online cell phone buyer reviews. Using various intuitive summary quality measures, we show that our method outperforms state-of-the-art review summarization methods (Sect. 5.3).

2 Problem Framework

Define an item (for example, a doctor or a camera) as a set of reviews, where each review is a set of *concept-sentiment* pairs $\{(c_1, s_1), (c_2, s_2), \ldots, (c_n, s_n)\}$, and $s_j \in \mathbb{R}$ is the sentiment for concept c_j in the review. We shows how to extract the concepts and their sentiments from the text of the reviews in Sect. 4.1, and Related Work (Sect. 6). The set of concepts are related based on a hierarchical *ontology* such as WordNet [19] and ConceptNet [23]. For instance, the "part-whole" relation in those ontologies can be utilized to create the hierarchy of aspects suitable for our framework. Alternatively, Kim et al. [12] automatically extract an aspect-sentiment hierarchy using a Bayesian non-parametric model.

We define the (directed) *distance* $d(p_1, p_2)$ between two concept-sentiment pairs $p_1 = (c_1, s_1)$ and $p_2 = (c_2, s_2)$, based on the concepts' relationship in the hierarchy, as follows.

Definition 1. *The distance $d(p_1, p_2)$ is:*

$$d(p_1, p_2) = \begin{cases} d(r, c_2) & \textit{if } c_1 \textit{ is the root } r, \textit{ or} \\ d(c_1, c_2) & \textit{if } c_1 \textit{ is the ancestor of } c_2 \textit{ and } |s_1 - s_2| \leq \epsilon, \textit{ or} \\ \infty & \textit{otherwise} \end{cases}$$

where the concept distance $d(c_1, c_2)$ is the shortest-path length from c_1 to c_2 in the hierarchy, r is the root of the hierarchy, and $\epsilon > 0$ is the sentiment threshold.

If pair p_1 has finite distance to p_2, we say that p_1 *covers* p_2. Pair p_1 covers p_2 iff p_1's concept c_1 is an ancestor of p_2's concept c_2, and either c_1 is the root concept or the sentiments of p_1 and p_2 differ by at most ϵ. Figure 1 shows an example of how the concept-sentiment pairs of an item's reviews are mapped on the concept hierarchy, where the dashed line is the path from the root, and concept c_6 doesn't have any pairs. For instance, pair $(c_1, 0.7)$ represents an occurrence of concept c_1 in a review with sentiment 0.7. The same pair is also represented by the circled 0.7 value inside the c_1 tree node.

Given a set $P = \{p_1, p_2, \ldots, p_q\}$ of concept-sentiment pairs for the reviews of an item, and an integer k, our goal is to compute a set $F = \{f_1, f_2, \ldots, f_k\} \subseteq P$ of k pairs that best summarize P. To measure the quality of such a summary F, we define its cost $C(F, P)$ as the distance from F to P, defined as follows.

Definition 2. *The distance from F to a pair p is the distance of the closest pair in $F \bigcup \{r\}$ to p: $d(F, p) = \min_{f \in F \bigcup \{r\}} d(f, p)$. The cost of F is the sum of its distances to pairs in P: $C(F, P) = \sum_{p \in P} d(F, p)$.*

We introduce two summarization problems as following:

1. ***k-Pairs Coverage:*** given a set P of concept-sentiment pairs (coming from a given set of reviews for an item) and integer $k \leq |P|$, find a subset $F \subseteq P$ with $|F| = k$ that summarizes P with minimum cost: $\min_{F \subseteq P, |F| = k} C(F, P)$
2. ***k-Reviews/Sentences Coverage:*** given a set R of reviews (or sentences) and integer $k \leq |R|$, find a subset $X \subseteq R$ with $|X| = k$ that summarizes

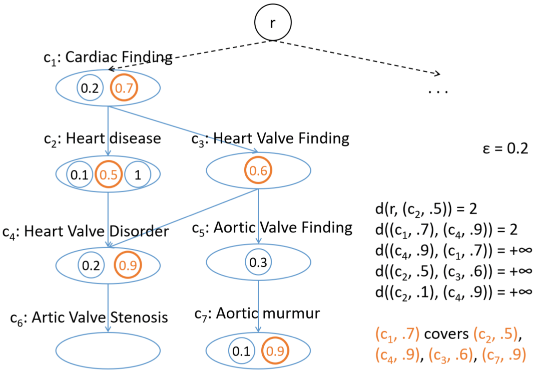

r
c_1: Cardiac Finding
0.2
0.7
. . .
c_2: Heart disease
c_3: Heart Valve Finding
ε = 0.2
0.1
0.5
1
0.6
c_4: Heart Valve Disorder
c_5: Aortic Valve Finding
0.2
0.9
0.3
d(r, (c_2, .5)) = 2
d((c_1, .7), (c_4, .9)) = 2
d((c_4, .9), (c_1, .7)) = +∞
d((c_2, .5), (c_3, .6)) = +∞
d((c_2, .1), (c_4, .9)) = +∞
c_6: Artic Valve Stenosis
c_7: Aortic murmur
0.1
0.9
(c_1, .7) covers (c_2, .5), (c_4, .9), (c_3, .6), (c_7, .9)

Fig. 1. Representation of concept-sentiment pairs on SNOMED-CT concept hierarchy

R with minimum cost: $\min_{X \subseteq R, |X|=k} C(P(X), P(R))$, where $P(R)$ is the set of concept-sentiment pairs derived from the set R of reviews/sentences, and $P(X)$ is the set of concept-sentiment pairs derived from the subset X of R.

Intuitively, the first problem is appropriate when the summaries consist of concise concept-sentiment pairs, e.g. “good Heart Disease management”, extracted from the reviews, and may be more suitable for mobile phone-sized screens. The second problem is appropriate if the summaries consist of whole sentences of reviews, which better preserves the meaning of the review, but may require more space to display.

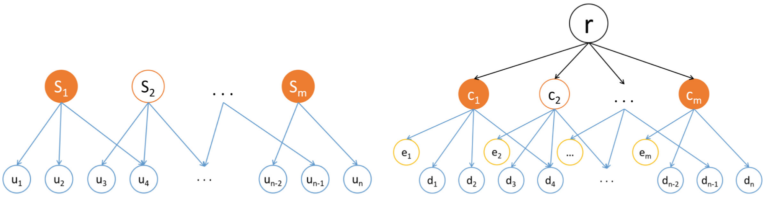
The k-Pairs Coverage problem can be viewed as a special case of the k-Reviews/Sentences Coverage problem, when each review/sentence has just one pair. For presentation simplicity, we first present our NP-hard proof and algorithms for k-Pairs Coverage in Sect. 4, then describe how they can be applied to the k-Reviews/Sentences Coverage in Sect. 4.5.

3 Both Problems are NP-Hard

This section proves both proposed problems NP-hard.

Theorem 1. *The k-Pairs Coverage problem is NP-hard.*

Proof. The decision problem is, given a set P of concept-sentiment pairs, an integer $k \leq |P|$, and a target $t \geq 0$, to determine whether there exists a subset $F \subseteq P$ of size k with cost $C(F, P)$ at most t. We reduce Set Cover to it. Fix any Set-Cover instance (S, U, k) where U is the universe $\{u_1, u_2, \ldots, u_n\}$, and $S = \{S_1, S_2, \ldots, S_m\}$ is a collection of subsets of U, and $k \leq |S|$. Given (S, U, k), first construct a concept-hierarchy (DAG) with root r, concepts c_i and e_i for each subset S_i, and a concept d_j for each element u_j. For each set S_i, make c_i a child of r and e_i a child of c_i. For each element u_j, make d_j a child of c_i for



(a) Set Cover (b) Converted to k-Pairs Coverage

Fig. 2. Reduction from set cover

each set S_i containing u_j. (See Fig. 2.) Next, construct $2m+n$ concept-sentiment pairs $P = \{p_1, \ldots, p_{2m+n}\}$, one containing each node in the DAG other than the root r, and all with the same sentiment, say 0. Take target $t = 3m + n - 2k$. This completes the reduction. It is clearly polynomial time. Next we verify that it is correct. For brevity, identify each pair with its node.

Suppose S has a set cover of size k. For the summary $F \subseteq P$ of size k, take the k concepts in P that correspond to the sets in the cover. Then each d_i has distance 1 to F, contributing n to the cost. For each set in the cover, the corresponding c_i and e_i have distance 0 and 1 to F, contributing k to the cost. For each set not in the cover, the corresponding c_i and e_i have distance 1 and 2 to F, contributing $3(m - k)$ to the cost, for a total cost of $n + 3m - 2k = t$.

Conversely, suppose P has a summary of size k and cost $t = n + 3m - 2k$. Among size-k summaries of cost at most t, let F be one with a maximum number of c_i nodes. We show that the sets corresponding to the (at most k) c_i nodes in F form a set cover. Assume some $c_{i'}$ is missing from F (otherwise $k \geq m$ so we are done). For every e_i in F, its parent c_i is also in F. (Otherwise adding c_i to F and removing e_i would give a better summary F', i.e., a size-k summary of cost at most t, but with more c_i nodes than F, contradicting the choice of F). No e_i is in F (otherwise removing e_i and adding the missing node $c_{i'}$ would give a better summary F'). No d_j is in F (otherwise, since neither $e_{i'}$ nor $c_{i'}$ are in F, removing d_j from F and adding $c_{i'}$ would give a better summary F'). Since no e_i or d_j is in F, only c_i nodes are in F. Since the cost is at most $t = n+3m-2k$, by calculation as in the preceding paragraph, the sets S_i corresponding to the nodes c_i in F must form a set cover. □

When we already have k-Pairs Coverage as a NP-hard problem, it's natural to prove the following theorem.

Theorem 2. *The k-Reviews/Sentences Coverage problem is NP-hard.*

Proof. K-Reviews/Sentences Coverage is a generalization of k-Pairs Coverage, so the theorem follows from the previous theorem.

4 Algorithms

We implement three algorithms for k-Pairs Coverage. The first, which is the only one generates an optimal solution, solves the standard integer-linear program

(ILP) for the problem, as a special case of the well-known k-Medians problem. The second randomly solves the linear program (LP), then randomly rounds the fractional solution achieving a bounded approximation error. The third is a greedy bounded approximation algorithm. The three algorithms share a common initialization phase that we describe first.

4.1 Initialization

The initialization phase computes the underlying edge-weighted bipartite graph $G = (U, W, E)$ where vertex sets U and W are the concept-sentiment pairs in the given set P, edge set E is $\{(p, p') \in U \times W : d(p, p') < \infty\}$, and edge (p, p') has weight equal to the pair distance $d(p, p')$. The initialization phase builds G in two passes over P. The first pass puts the pairs $p = (c, s)$ into buckets by category c. The second pass, for each pair $p = (c, s)$, iterates over the ancestors of c in the DAG (using depth-first-search from c). For each ancestor c', it checks the pairs $p' = (c', s')$ in the bucket for c'. For those with finite distance $d(p, p')$, it adds the corresponding edge to G.

For our problems, the time for the initialization phase and the size of the resulting graph G are roughly linear in $|P|$, because the average number of ancestors for each node in the DAG is small.

4.2 ILP for Optimal Solution

Given the graph $G = (U, W, E)$, we adapt the standard k-Medians ILP for our non-standard cost function as below.

$$\begin{array}{ll} \text{minimize} & \sum_{(p,q)\in E} y_{pq} \times d(p,q) \\ \text{subject to} & x_r = 1; \qquad \sum_{p\in P\setminus\{r\}} x_p = k; \qquad \sum_{\forall q\in W, p:(p,q)\in E} y_{pq} = 1 \\ & (\forall (p,q) \in E \quad 0 \le y_{pq} \le x_p; \qquad (\forall p \in U) \quad x_p \in \{0,1\} \end{array}$$

Our first algorithm solves the ILP using the Gurobi solver. Of course, no worst-case polynomial-time bounds are known for solving this NP-hard ILP, but on our instances the algorithm finishes in reasonable time (Details are in Sect. 5).

4.3 Randomized Rounding

The second algorithm computes an optimal fractional solution (x, y) to the LP relaxation of the ILP (using Gurobi, details in Sect. 5), then randomly rounds it as shown in Algorithm 1: it chooses the summary F by sampling k pairs p at random from the distribution $x/\|x\|_1$. No good worst-case bounds are known on the time to solve the LP, but on our instances the solver solves it in reasonable time. The randomized-rounding phase can easily be implemented to run in linear time, $O(n)$ where $n = |P|$. This randomized-rounding algorithm is due to [27] (see also [4]). The following worst-case approximation guarantee holds for this algorithm, as a direct corollary of the analysis in [4]. Let $\text{OPT}_k(P)$ denote the minimum cost of any size-k summary of P.

Algorithm 1. Randomized Rounding Algorithm

```
Input: fractional solution $x, y$
Output: summary $F$
procedure RANDOMIZED ROUNDING
    Define probability distribution $q$ on $P' = P \setminus \{r\}$ such that $q(p) = \frac{x_p}{\sum_{p \in P'} x_p}$.
    $F = \emptyset$
    while $|F| < k$ do
        Sample one pair $p$ without replacement from $q$.
        Add $p$ to $F$.
    Return $F$.
```

Theorem 3. *The expected cost of the size-k summary returned by the randomized-rounding algorithm is $O(\text{OPT}_{k'}(P))$ for some $k' = O(k/\log n)$.*

In our experiments it gives near-optimal summary costs.

4.4 Greedy Algorithm

The greedy algorithm is Algorithm 2. It starts with a set $F = \{r\}$ containing just the root. It then iterates k times, in each iteration adding a pair $p \in P$ to F chosen to minimize the resulting cost $C(F \cup \{p\}, P)$. Finally, it returns summary $F \setminus \{r\}$. This is essentially a standard greedy algorithm for k-medians. Since the cost is a submodular function of P, the algorithm is a special case of Wolsey's generalization of the greedy set-cover algorithm [26].

After the initialization phase, which computes the graph $G = (U, W, E)$, the algorithm further initializes a max-heap for selecting p in each iteration. The max-heap stores each pair p, keyed by $\delta(p, F) = C(F \cup \{p\}, P) - C(F, P)$. The max-heap is initialized naively, in time $O(m + n \log n)$ (where $m = |E|$, $n = |P|$). (This could be reduced to $O(m + n)$ with the linear-time build-heap operation.) Each iteration deletes the pair p with maximum key from the heap (in $O(\log n)$ time), adds p to F, and then updates the changed keys. The pairs q whose keys change are those that are neighbors of neighbors of p in G. The number of these updates is typically $O(d^2)$, where d is the typical degree of a node in G. The cost of each update is $O(\log n)$ time. After initialization, the algorithm typically takes $O(kd^2 \log n)$ time. In our experiments, our graphs are sparse (a typical node p has only hundreds of such pairs q), and k is a small constant, so the time after initialization is dominated by the time for initialization. The following worst-case approximation guarantee is a direct corollary of Wolsey's analysis [26]. Let $H(i) = 1 + 1/2 + \cdots + 1/i \approx 1 + \log i$ be the ith harmonic number. Let Δ be the maximum depth of the concept DAG.

Theorem 4. *The greedy algorithm produces a size-k summary of cost at most $\text{OPT}_{k'}(P)$, where $k' = \lfloor k/H(\Delta n) \rfloor$.*

In our experiments, the algorithm returns near-optimal size-k summaries.

Algorithm 2. Greedy Algorithm

```
Input: $G = (U, W, E)$ from initialization, computed from $P$.
Output: Size-$k$ summary $F$
procedure GREEDY
    Define $\delta(p, F) = C(F \cup \{p\}, P) - C(F, P)$.
    Initialize $F = \{r\}$, and max-heap holding $p \in U$ keyed by $\delta(p, F)$.
    while $|F| < k + 1$ do
        Delete $p$ with highest key from max-heap.
        Add $p$ to $F$.
        for $w$ such that $(p, w) \in E$ do
            for $q$ such that $(q, w) \in E$ do
                Update max-heap key $\delta(q, F)$ for $q$.
    return $F \setminus \{r\}$
```

4.5 Adaptation for k-Reviews/Sentences Coverage Problem

When whole reviews or sentences (each containing a set of concept-sentiment pairs) must be selected, the above algorithms can still be applied with only a modification of the initialization stage. In particular, we modify the construction of bipartite graph $G = (U, W, E)$, so instead of having both U and W be concept-sentiment pairs in P, U represents the set of candidate reviews or sentences R, and W represents concept-sentiment pairs as before. Therefore the edge set E becomes $\{(r, p) \in U \times W : d(r, p) < \infty\}$, and edge (r, p) has weight equal to the distance $d(r, p)$ from review/sentence r to pair p. After this initialization, the algorithms work as usual.

5 Experimental Evaluation

In this section we conduct both quantitative and qualitative evaluations. The quantitative evaluation measures the time and accuracy trade-offs of the proposed approximate summarization algorithms compared to the optimal solution. The qualitative evaluation evaluates the quality of the summaries generated by the proposed methods, compared to baseline state-of-the-art unsupervised summarization methods using several intuitive measures.

5.1 Experiment Setup

Datasets: We utilize two real-world datasets: health care and online consumer reviews. Our first dataset consists of 68,686 patient reviews of the 1000 most reviewed doctors from vitals.com, which is a popular doctor rating website. As the second dataset, we crawled customer reviews of 60 unlocked cell phones, which are featured in the first five pages on Amazon and have at least 100 distinct reviews each. Table 1 presents basic statistics of the two datasets.

Table 1. Dataset characteristics.

	Doctor reviews	Cell phone reviews
#Items (doctor/product)	1000	60
#Reviews	68686	33578
Min #reviews per item	43	102
Max #reviews per item	354	3200
Average #sentences per review	4.87	3.81

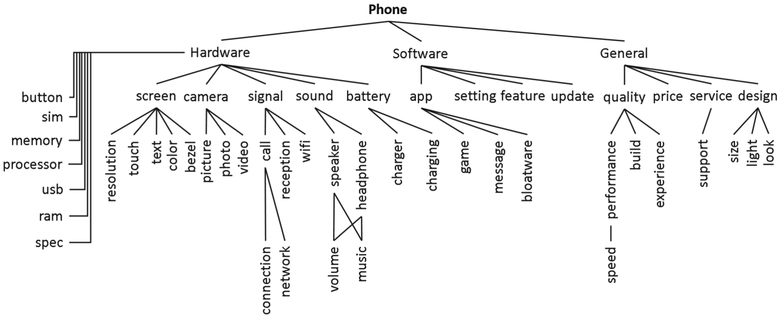
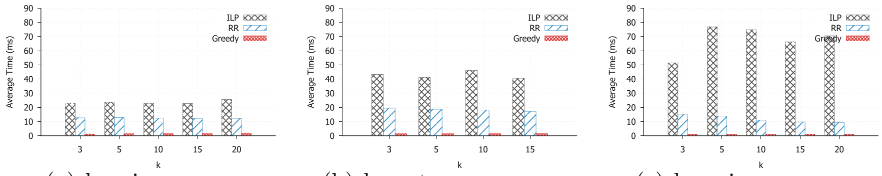


Fig. 3. Cell phone aspect hierarchy

Concepts and Sentences Extraction: To extract medical concepts in doctor reviews we use automated tool MetaMap [1] and SNOMED CT [10] ontology, which has more than 300,000 concepts and is suitable for our problem given its focus on describing medical conditions. For example, for sentence *"Dr Robert did an awesome job with my tummy tuck and liposuction"*, concepts *"tummy tuck"* (UMLS ID = C0749734) and *"liposuction"* (ID = C0038640) are extracted. In cell phone reviews dataset, we employ Double Propagation method [22] to extract aspects such as screen and battery. We only focus on the 100 most popular extracted aspects. Since there is no available hierarchy of cell phone aspects, we manually built a hierarchy from the extracted aspects as shown in Fig. 3.

Sentiment Computation: To compute the sentiment around a concept, we compute the sentiment of the containing sentence and assign this sentiment to the concept. We adopt a neural network based representation learning approach *doc2vec* to represent sentences by fixed-size vectors [15]. Then, sentence's sentiment estimation is formulated as a standard regression problem using the sentence vector representation.

Configuration: We evaluate the three methods proposed in Sect. 4: Integer Linear Programming - ILP, Randomized Rounding - RR, and Greedy algorithm. For ILP and RR, we use the Gurobi optimization library version 6.0.5 [8] with Dual-Simplex as the default method. This method is chosen because it shows the best performance in our case after experimental trials on different options



(a) k-pairs coverage (b) k-sentences coverage (c) k-reviews coverage

Fig. 4. Time evaluation with threshold 0.5

available in Gurobi (primal simplex, barrier, auto-switching between methods, concurrent). All experiments were executed on a single machine with Intel i7-4790 3.60 GHz, 16 GB RAM, Windows 10. Our code was written in Java 8.

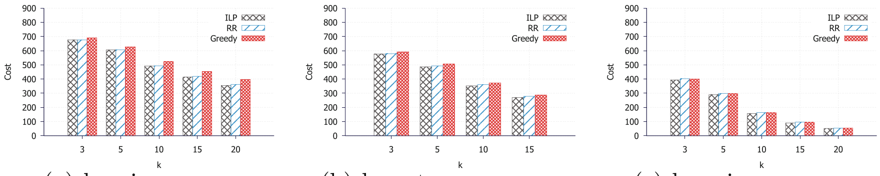
5.2 Quantitative Evaluation

For brevity, we only present results on doctor reviews dataset, which is the larger dataset, in this section. We compare the average coverage cost (defined in Definition 2) and time of our three algorithms. Due to space limitation we only present results with threshold (ϵ) 0.5 in Figs. 4 and 5, while results of other thresholds show similar trends.

A key observation from these experiments is that Greedy is always the fastest algorithm while maintaining reasonable costs compared to ILP and RR. Of course, ILP gives optimal solution, thus offers the cheapest cost. The Greedy algorithm has the worst cost but never more than 8% higher than the optimal ($\leq$ 5% most of time). In terms of time, the Greedy outperforms ILP by a factor up to 19×, 32× and 63× in the top pairs, top sentences and top reviews problems, respectively. Similarly, Greedy runs faster than RR, at most 14 times, and usually takes only 1–2 ms per doctor. RR algorithm works similarly to ILP regarding cost, specifically, the difference is about 1–2%. The speedup of RR over ILP is about 2–5×. This is because RR only solves a Linear Program system and then randomizes the solution instead of finding an optimal integer solution.

We also notice that with the same threshold, the cost decreases from top pairs to top sentences, and then to top reviews problem. The reason is that a sentence or review can have multiple pairs, so they typically cover more pairs than a single pair can cover. Therefore, k sentences or reviews usually cover more pairs than k pairs can, which leads to smaller costs. Similarly, the elapsed time of all algorithms for top sentences/reviews problem are larger than for top pairs problem. It's because for top sentences/reviews, there are more connections (edges) between selecting candidates and pairs to consider.

In general, the results suggest that our problem has latent structures friendly to Greedy algorithm. Therefore, the optimal solution from ILP algorithm seem to be close to the one of Greedy algorithm which can be achieved much faster. Therefore, we choose Greedy algorithm for the next qualitative experiments.



(a) k-pairs coverage (b) k-sentences coverage (c) k-reviews coverage

Fig. 5. Cost evaluation with threshold 0.5

5.3 Qualitative Evaluation

The goal of this section is to study the quality of the summarization achieved by the proposed algorithms, compared to several state-of-the-art unsupervised baselines. We focus on the sentence selection problem variant, which offers a balance between conciseness and semantic completeness.

Selecting Sentiment Threshold ϵ Used by Greedy Algorithm: We select the threshold value ϵ for which the rate of covered sentences significantly drops if we further increase ϵ. For that, we employ the elbow method, which shows that the sentiment threshold's elbow is at 0.5 most of time (details removed for brevity). Intuitively, this sentiment threshold is also reasonable in the sense that a very positive sentiment of value 1.0 can cover a positive sentiment of value 0.5. Therefore, we choose sentiment threshold 0.5 for our greedy summarizer.

Baseline Summarization Methods: Our baselines come from two areas: one from opinion summarization approach, and the other from multi-document summarization. Specifically, the first baseline method to select top k sentences is adapted from Hu et al. [9]. This algorithm was designed to summarize customer reviews of online shopping products. It first extracts product aspects (attributes like "picture quality" for product "digital camera"), then classifies review sentences that mention these aspects as positive or negative, and finally sums up the number of positive and negative sentences for each aspect. To have a fair comparison, we adapt their method to select top k sentences into summaries. We first count the number of pair (concept, positive) or (concept, negative), for example: aspect "picture quality" with sentiment "positive" occurs in 200 sentences. Then, we select k most popular pairs and return one containing sentence for each selected pair. Note that the aspect extraction task is common in both the baseline and our methods. We refer to this baseline as *"most_popular"* since their summarizer favors the most popular aspects.

The second baseline from the opinion summarization area is adapted from a review summarizer [3] of local services (such as hotels, restaurants). This method selects the (aspect, positive/negative) pairs proportionally to the pair's frequency instead of selecting the most popular pair as in *"most_popular"* method. Then, it pick the new, most extremely polarized sentence to represent each selected pair (concept, positive/negative). We name this summarizer as *"proportional"*.

The other set of baselines are popular extractive, unsupervised multi-document summarizers that are agnostic to a concept's sentiment orientation. Contrasting to abstractive summarizers that compose summaries by creating brand-new sentences, extractive summarizers make use of original documents' sentences, hence it is appropriate to be compared with our method. TextRank [18] summarizer applies PageRank algorithm on text by modelling text as graph of sentences in which sentences' similarity is considered as sentence-to-sentence edge weight. LexRank [6] is another document summarizer relying on a sentence graph for detecting the most important sentences. The last baseline in this line is Latent Topic Modelling (LSA) based summarizer [24], which utilizes the sentence's vector representation calculated using Singular Value Decomposition (SVD) on a term-sentence matrix. In our experiments, We utilize Sumy [2] library for these three methods. We summarize all baselines with brief descriptions in Table 2.

Table 2. Baseline unsupervised summarizers

Most_popular	[9]	Pick representative sentences of popular aspect-polarity pairs
Proportional	[3]	Pick representative sentences with extreme sentiments after selecting aspects proportionally
TextRank	[18]	No sentiment, use sentence graph with word overlap for sentence similarity
LexRank	[6]	No sentiment, use sentence graph with cosine-based sentence similarity
LSA-based	[24]	No sentiment, utilize SVD on term-sentence matrix

Summary Quality Measures and Results: We evaluate all methods on a new measure, named *"sentiment error"* (or *"sent-err"*), to avoid giving an unfair advantage to our method. Note that typical multi-document summarization measures such as ROGUE are not applicable in our context since they do not consider sentiment and concept relationship. The key idea is to look at the difference between every concept's sentiment in the original reviews and that concept's sentiment (extrapolated if concept not in summary) in the summary. That is, for each pair in the original reviews, we find the closest concept in the summary and measure the *sentiment distance* between them. In contrast, in Definition 2, we measure the *concept distance* (in hierarchy edges) between a review concept and its nearest covering summary concept.

Recall that we summarize a set of concept-sentiment pairs P by a subset F contained in k sentences. We define *"sent-err"* of F with respect to P in a root-mean-square error manner: $\mathit{sent\text{-}err}(P, F) = \sqrt{\frac{1}{|P|}\sum_{p \in P} \mathit{err}_{p,F}^2}$, where p is a pair of (concept c_p, sentiment s_p). $\mathit{err}_{p,F}$ (Eq. 1) is the smallest difference between s_p and that concept's sentiments in a pair in F. When concept c_p does not appear in F, we use the sentiments of c_p's lowest ancestor in F if available.

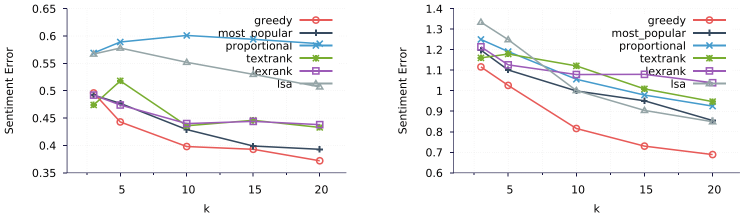
When neither c_p nor its ancestors appear in F, we consider a neutral sentiment 0. The intuition is that the error models the difference of every concept's sentiment and the closest sentiment of that concept or its ancestors in summary.

$$err_{p,F} = \begin{cases} \min_{f \in F, c_f = c_p} |s_f - s_p| & : c_p \in F \\ \min_{f \in F, c_f = c_p\text{'s ancestor}} |s_f - s_p| & : c_p \notin F \land c_p\text{'s ancestor} \in F \\ |0 - s_p| = |s_p| & : \text{otherwise} \end{cases} \quad (1)$$

Another version of this measure penalizes the case of missing concept c_p and its ancestor in summary F by considering the largest possible error of c_p's sentiment. In another words, the third branch of Eq. (1) becomes $err_{p,F} = \max(|1 - s_p|, |-1 - s_p|)$. Note that $+1$ and -1 are the extreme sentiments in our model. We name this measure version as *"sent-err-penalized"*.

Results: Figure 6 compare the errors of our method and the baselines on cell phone review dataset (similar results on doctor reviews dataset). On the first measure, *sent-err* (Fig. 6(a)), we find that our method always leads to the smallest sentiment error, i.e. highest-quality summaries. It can reduce the error of the second best performance method ("most_popular") by 4.1% on average, and other methods by 14.6% on average. The multi-document summarization methods generally perform poorly since they ignore the sentiment. Our method reduces those multi-document summarizers' error by up to 23.7%. The errors of all methods drop when the number of summary sentences increases, as expected.

On *sent-err-penalized* measure (Fig. 6(b)), our method beats all baselines with larger margins. Specifically, our method improves the error of second best performance method ("most_popular") 14.9%, and other methods by 19.8% on average. This result indicates that missing concepts in summary problem is more severe in baseline methods, and our method is smarter in choosing sentences.



(a) sent-err measure

(b) sent-err-penalized measure

Fig. 6. Sentiment error on cell phone reviews dataset (lower error is better)

6 Related Work

Multi-document Summarization: This is a traditional problem with the most well-known applications in summarizing online news articles.

Goldstein et al. presented a typical method [7], which extend single-document summarization techniques. A key difference is that there is more redundancy across documents of a similar topic than within a single document. This is an observation we also adopt in our work. TexRank [18] and LexRank [6] are two popular, similar methods based on building weighted graph of document sentences, which are rated by Pagerank algorithm to pick the important ones. Steinberger et al. [24] proposed an LSA-based summarizer that utilizes sentence's vector representation in their latent index space. Recent deep learning based approaches [11,16] are supervised and/or abstractive summarizers while our summarization method is unsupervised and extractive. However, none of above methods consider the sentiment in input documents. We incorporate some of these methods (TextRank, LexRank and LSA-based) as baselines in our evaluations (Sect. 5).

Sentiment Analysis: The methods fall into two categories, using unsupervised or supervised learning. The unsupervised methods [25] focus on building a comprehensive opinion word dictionary, or use linguistic rules to find opinion phrases containing adjectives or adverbs in a document. An early supervised method [21] applies a Bag-Of-Word model to classify movie reviews as positive or negative. Recently, a common approach [15] is to use neural network model to extract the better review's vectors, thus get the better results on sentiment classification task. Any of these methods can be plugged into our framework.

Aspect Extraction: A common review analysis task is to extract the product aspects. Traditional methods [9,22] use association mining to find frequent aspects, then apply pruning rule to remove meaningless, redundant ones; later they also have a rule to discover additional infrequent aspects based on both frequent ones and opinion words. A semi-supervised approach based on topic modelling extract product aspects as multi-grain topics [20]. Extracting aspects is outside the scope of this paper. We use Metamap [1] and Double Propagation technique [22] in our experiments.

Opinion Summarization: The most popular approach is based on aspect extraction. Hu et al. [9] first extract product aspects from online customer reviews, then report the number of positive/negative sentences for each aspect. This can be augmented by showing aggregated rating along with representative phrases [17], or sentences [3] for each aspect. Different from this kind of statistical summaries, Lappas et al. [13] formulates the problem as selecting k reviews that optimize the aspect coverage while rewarding high-quality reviews, or maintaining their proportion of aspect opinions.

A key difference of this paper from all the above works is that they do not consider the relationships between the aspects nor a continuous sentiment scale. A preliminary version of this work, which focuses on coverage measure of the greedy algorithm on the doctor reviews dataset, was published as a poster [14].

7 Conclusions

We introduced a novel review summarization problem that considers both the ontological relationships between the review concepts and their sentiments. We described methods for extracting concepts and estimating their sentiment. We proved that the summarization problem is NP-hard even when the concept ontology is a DAG, and for that we presented efficient approximation algorithms We evaluated the proposed methods extensively with both quantitative and qualitative experiments. We found that the Greedy algorithm can achieve quality comparable to the optimal is much shorter time, comparing to other algorithms. Moreover, using various coverage measures and sentiment error measures, we show that the Greedy outperforms a baseline method on selecting k sentences to summarize real reviews.

Acknowledgment. This work was partially supported by NSF grants IIS-1838222, IIS-1619463, IIS-1901379 and IIS-1447826

References

1. Aronson, A.R.: Effective mapping of biomedical text to the UMLS metathesaurus: the metamap program. In: AMIA (2001)
2. Belica, M.: Sumy: Module for automatic summarization of text documents (2017)
3. Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G.A., Reynar, J.: Building a sentiment summarizer for local service reviews. In: WWW Workshop on NLP in the Information Explosion Era (2008)
4. Chrobak, M., Kenyon, C., Noga, J., Young, N.E.: Oblivious medians via online bidding. In: Correa, J.R., Hevia, A., Kiwi, M. (eds.) LATIN 2006. LNCS, vol. 3887, pp. 311–322. Springer, Heidelberg (2006). https://doi.org/10.1007/11682462_31
5. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: WSDM 2008. ACM (2008)
6. Erkan, G., Radev, D.R.: LexRank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. **22**, 457–479 (2004)
7. Goldstein, J., Mittal, V., Carbonell, J., Kantrowitz, M.: Multi-document summarization by sentence extraction. In: NAACL-ANLP (2000)
8. Gurobi Optimization: Gurobi optimizer manual (2015). http://gurobi.com
9. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD (2004)
10. International: SNOMED CT (2016). https://www.snomed.org/snomed-ct
11. Jadhav, A., Rajan, V.: Extractive summarization with swap-net: sentences and words from alternating pointer networks. In: ACL (2018)
12. Kim, S., Zhang, J., Chen, Z., Oh, A.H., Liu, S.: A hierarchical aspect-sentiment model for online reviews. In: AAAI (2013)
13. Lappas, T., Crovella, M., Terzi, E.: Selecting a characteristic set of reviews. In: KDD (2012)
14. Le, N.X., Hristidis, V., Young, N.: Ontology- and sentiment-aware review summarization. In: ICDE, pp. 171–174 (2017)
15. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: ICML (2014)

16. Liu, L., Lu, Y., Yang, M., Qu, Q., Zhu, J., Li, H.: Generative adversarial network for abstractive text summarization. In: AAAI (2017)
17. Lu, Y., Zhai, C., Sundaresan, N.: Rated aspect summarization of short comments. In: World Wide Web 2009 (2009)
18. Mihalcea, R., Tarau, P.: TextRank: bringing order into text. In: EMNLP (2004)
19. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
20. Mukherjee, A., Liu, B.: Aspect extraction through semi-supervised modeling. In: 50th Annual Meeting of ACL (2012)
21. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: ACL-2002 Conference on EMNLP (2002)
22. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Comput. Linguist. **37**(1), 9–27 (2011)
23. Speer, R., Havasi, C.: Representing general relational knowledge in conceptNet 5. In: LREC (2012)
24. Steinberger, J., Jezek, K.: Using latent semantic analysis in text summarization and summary evaluation. In: Proceedings of ISIM 2004 (2004)
25. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Comput. Linguist. **37**(2), 267–307 (2011)
26. Wolsey, L.A.: An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica **2**(4), 385–393 (1982)
27. Young, N.E.: K-medians, facility location, and the Chernoff-Wald bound. arXiv preprint arXiv:cs/0205047 (2002)