Generating Comparative Summaries from Reviews

Ruben Sipos
Dept. of Computer Science
Cornell University
Ithaca, NY 14853 USA
rs@cs.cornell.edu

Thorsten Joachims
Dept. of Computer Science
Cornell University
Ithaca, NY 14853 USA
tj@cs.cornell.edu

ABSTRACT

To facilitate direct comparisons between different products, we present an approach to constructing short and comparative summaries based on product reviews. In particular, the user can view automatically aligned pairs of snippets describing reviewers' opinions on different features (also selected automatically by our approach) for two selected products. We propose a submodular objective function that avoids redundancy, that is efficient to optimize, and that aligns the snippets into pairs. Snippets are chosen from product reviews and thus easy to obtain. In our experiments, we show that the method constructs qualitatively good summaries, and that it can be tuned via supervised learning.

Categories and Subject Descriptors

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

Keywords

summarization; reviews; comparison; snippets; pairs; submodular

1. INTRODUCTION

After deciding what kind of product to buy (e.g. a cell phone), it is often still hard to make a good choice due to abundance of different brands and models. Using the internet as a convenient source of information, we can usually find a wealth of data describing many products – even more so if we plan to make our purchase online too. First, reading in-depth professional reviews might help us to familiarize ourselves with strengths and weaknesses of a certain product, but does not directly give us direct understanding of how it compares to other choices. Second, we can find tables comparing products, but they usually list only product feature specifications as documented by the manufacturer and do not provide insight into how those features translate into usefulness during the actual use. Third, we can find user reviews of products which describe experiences of using the product and are often helpful in making the purchasing decision. However, the major drawbacks are the need to read many reviews to get an estimate of agreement between users and the need to manually compare different products (by keeping the contents of the reviews in mind).

Our aim is to fill-in this space and provide a way of explicitly comparing products of the same type (e.g., competing cell phones). These comparisons should reflect users' experiences (i.e., are based on product reviews), so that they can complement specification-based comparisons tables.

Many online retailer websites already provide user reviews. In our approach, we offer an additional view of these reviews by providing a compact (to avoid the need to read many long reviews) and comparative (to facilitate decision support) summary of them. In other words, we select important snippets from the reviews of one product and present them aligned with snippets talking about the same aspect from the reviews of the other product. This allows users to quickly read through the aligned pairs describing reviewers' opinions on important aspects of the two products they are trying to compare.

Product	Comparative Snippet Pairs
A	battery lasted for about 7h of web browsing
B	I got about 8h but only if I disabled wireless
A	screen has a good uniform lighting
B	there was a slight light bleed on the screen
A	buttons were hard to press
B	despite small buttons, they were easy to use

Figure 1: Illustrative example of a comparative summary for products A and B.

Figure 1 shows an example of what we would expect from the system. Each pair talks about one aspect of the product and the selection should represent the most important ones (as defined by the reviewers' selection of which ones to mention). The snippets give insight into how well the product specifications translate into real-world utility based on reviewers' experiences and how do they compare to a competing product. Reading such summary provides an alternative to time-consuming reading of many reviews and at the same time provides direct comparison between different purchasing choices.

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/10.1145/2505515.2507879.

2. MODEL

We assume that the user selects two products A & B to compare against each other, and that each product $p \in \{A, B\}$ comes with a respective set of reviews $R^p = r_1^p, ..., r_{n^p}^p$ (obtained from e.g. online retailer's website). Each set of reviews R^p is then split into snippets $S^p = s_1^p, ..., s_{m^p}^p$ (by e.g. using sentences as snippets). Finally, we use our objective function F to select a set of snippet pairs (s_i^A, s_j^B) (where both snippets describe the same aspect but for two different products) that represent the final summary.

Each snippet is represented as a vector (in the simplest case using a bag-of-words TFIDF scores; otherwise snippets can be represented using any features) and $v_k^p(x)$ represents the value of feature x in snippet s_k^p (e.g. TFIDF score of word x). To construct a pair, we want to select two snippets (one for each product) such that they talk about the same aspect (i.e. are aligned). To promote good alignment we use sum of features in the intersection of the two snippets (Eq. 1). Furthermore, we want to also account for the information outside the intersection to promote showing the differences (instead of just finding the most similar snippets). To achieve this we use the sum of features outside of the intersection (Eq. 2), but clamped to be less or equal to the value of the intersection. Clamping the score avoids assigning high scores to some bad corner-cases (e.g. two long snippets being aligned only on the word "the"). Finally, for any candidate ($s_i^A \in S^A, s_j^B \in S^B$) we then use Eq. 3 (with the favorable properties of having a good alignment and including useful comparative information) to score the given pairing of snippets.

$$c(s_i^A, s_j^B) = \sum_{x \in s_i^A \wedge x \in s_j^B} v_i^A(x) \tag{1}$$

$$g(s_i^A, s_j^B) = \min\{\sum_{x \in s_i^A \wedge x \notin s_j^B} v_i^A(x), c(s_i^A, s_j^B)\} \tag{2}$$

$$\begin{aligned} f(s_i^A, s_j^B) = g(s_i^A, s_j^B) + c(s_i^A, s_j^B) + \\ c(s_j^B, s_i^A) + g(s_j^B, s_i^A) \end{aligned} \tag{3}$$

Let's consider some examples of bad snippet pairs to better understand why we selected this scoring function. Snippet pair *(the battery life is good, the screen had some light bleed)* is a bad choice because the snippets overlap only on the word "the". It does not get chosen because the score is limited by the intersection. Pair *(all buttons are big and very easy on the fingers, has buttons)* is highly unbalanced (left snippet adds a lot of weight if we naively summed the word weights) but the score is again limited by the intersection. In the third example *(and the screen has fantastic colors, while the screen has fantastic colors)* we have a large intersection, but still a low score because we do not have much product specific information (and thus parts outside the intersection do not add much weight). Our approach still includes coverage terms (and thus gives weight to frequently occurring phrases), however it prefers to select and point out the differences (to facilitate deciding between the products).

In contrast to scoring a single pair of snippets (Eq. 3), the main objective function (Eq. 7) is a submodular set function. In addition to selecting aligned pairs talking about important aspects, we can now select diverse pairs that give a good coverage of the source information (original reviews) while avoiding redundancy. We parametrize this objective with weight vector $\mathbf{w}$ (where scalar w_x corresponds to the feature x), which allows for supervised learning, but can be substituted with $\mathbf{1}$ in the simple case of *uniform weights*.

The objective function F (Eq. 7) follows a similar pattern to the individual pair scoring. To simplify the notation we use $\alpha(S)$ to represent the union of all words present in snippets for product A included in S but not for B, similarly $\beta(S)$ for words in B but not A and $\gamma(S)$ for the union of all words in the intersections. Function H sums the largest weights for all words (thus resulting in diminishing returns for covering a word multiple times) in the target set (i.e. intersection $X = \gamma(S)$ or the remainder corresponding to A or B). The final objective (Eq. 7) is composed of four parts (corresponding to intersections and remainders using weights for A or B).

$$H^A(S, X) = \sum_{x \in X} \max_{(i,j) \in S} w_x v_i^A(x) \tag{4}$$

$$H^B(S, X) = \sum_{x \in X} \max_{(i,j) \in S} w_x v_j^B(x) \tag{5}$$

$$G^p(S, X) = \min\{H^p(S, X), H^p(S, \gamma(S))\} \tag{6}$$

$$\begin{aligned} F(S) = G^A(S, \alpha(S)) + H^A(S, \gamma(S)) + \\ H^B(S, \gamma(S)) + G^B(S, \beta(S)) \end{aligned} \tag{7}$$

Eq. 7 can be efficiently maximized using a greedy algorithm (linear in the number of candidate pairs and selected set cardinality) which achieves a constant factor approximation [8] and works well in practice.

3. EXPERIMENTS

Data. To evaluate our approach we used reviews from Amazon's web site. We scraped reviews for 8 tablets (from different manufacturers) and split them into sentences based on punctuation. Then we parsed those sentences and, if necessary, further split them into smaller snippets (e.g. in the case of two clauses connected by "and"). By doing this we obtain in the order of 10000 snippets per product. Experiments were performed using TFIDF weighting, where we treat each review as one document.

Filtering. Because we are selecting pairs of snippets, the number of candidate pairs (and thus running time of the greedy algorithm) grows quadratically with the number of snippets obtained from the reviews. To speed up the selection of which pairs to present to the user, we precomputed the set of the top most similar snippets (using cosine similarity). For each product pair (for which we want to show a comparison) we limit ourselves to the top 10000 most similar pairs while running the greedy algorithm. We believe that this is a reasonable number based on our empirical observations and the fact that even within this limited set the similarities in the bottom part already became very weak.

Labels. For the purposes of the evaluation and supervised learning we require labeled data. Our labels are per snippet pair and are defined as follows:

+1.0 a good pair (the two snippets are talking about the same aspect and contain relevant/useful information about the product)

+0.5 a misaligned pair (at least one snippet contains useful information but it does not talk about the same thing as the other one, e.g. *"battery life was good"* and *"the battery is hard to replace"*)

-0.5 irrelevant comments (e.g. reviewer discussing seller's customer service response)

-1.0 a bad pair (the pair contains no useful information, is nonsensical, etc.)

Qualitative Evaluation. On Figure 2 we show the top 5 pairs selected by our method comparing the Apple iPad with the Google Nexus tablet. All selected pairs are "good" according to the labeling and they also fit our goals: they are aligned (both snippets talk about the same aspect, e.g. microsd slot), they are balanced (no very short snippets paired with extremely long ones), and they talk about a non-redundant set of topics (e.g. storage space, expansion slots, email accounts etc.).

Uniform weights. Our approach can be used in a non-learning setting (where we do not require labels except for the evaluation purposes). In this case we use *uniform weights* by setting $\mathbf{w} = \mathbf{1}$. We compare the results of this scoring function against the following *baseline*, which simply selects the top most similar snippets as the final output. The only additional restriction is that no snippet may be selected more than once. Note that this simple baseline does not account for redundancy (except for not repeating the same snippet) and coverage of what is important, but only strives to maximize the intersections of snippets.

Comparison of our method with the baseline for selecting 40 pairs is shown in Table 1. It demonstrates that the simple baseline results in substantially worse performance than using our approach with uniform weights. Furthermore, after manually comparing them we believe that the pairs selected by the baseline are qualitatively worse overall than the ones selected by our approach.

Learning. The goal of learning in our approach is to generalize across product pairs. For example, if we get some labels for comparison of camera models A and B, we would like to use this information to improve the performance for the comparisons of C and D as well. We can expect this to be possible because, for example, indicating that resolution is an important factor most likely applies across all models.

In our experiments for the supervised learning case, we simulate an online setting as we would expect it in a deployed version. For a given product pair that we want to compare, we select 5 pairs to be presented to the user using our approach. For each individual pair, we receive user feedback expressed as labels as defined in the *Labels* paragraph. After each such iteration, we retrain our model using all the labels obtained so far. This is easily doable due to small number of training examples, but one could also use an incremental learning approach. Furthermore, to facilitate exploration and to simplify the labeling process, we do not allow the same pair to be selected again in the following iterations. In this way we obtain 100 labels by doing 20 iterations of presenting 5 pairs (using weights computed in the previous iteration). Altogether, one annotator labeled 100 training snippet pairs for each of 4 distinct product pairs. For training we use a linear support vector regression (using the label values) on a bag-of-words representation (with TFIDF weighting).

Table 1: Average performance scores of *baseline*, using *uniform weights* and *learned weights* across product pairs. Our approach outperforms the baseline and achieves more than 20% improvement on previously unseen product pairs through learning.

method	score	standard error
baseline	7.6	4.2
uniform weights	17.3	0.9
learned weights	22.1	1.1

The performance is measured by selecting 40 pairs from a different product pair with a disjoint set of reviews (to avoid any possible overlap in the data) but using the same weights, and computing the score according to the labeling. The reason for selecting a larger set of pairs (40) is to obtain a more robust score, because measuring smaller performance changes on only 5 pairs is unreliable due to low granularity. The results are then averaged across all combinations of one training and one testing product pair. The comparison in Table 1 shows more than 20% increase in the performance above the uniformly weighted case (which is already good in itself by looking at the qualitative evaluation on Figure 2).

Model variants. We experimented with other possible features in addition to bag-of-words TFIDF scores, but the ones we tried did not noticeably improve the score (which is already high for the basic model). Furthermore, minor changes to the scoring function do not immediately break the model from what we observed. Also, selecting pairs individually instead of using the global submodular objective still produces reasonable results, but introduces noticeable amounts of redundancy into the summary as expected.

4. RELATED WORK

Instead of performing generic summarization [2] or selecting one representative sentence [1] based on a single corpus, we are constructing summaries based on two sets of documents (in our case snippets from reviews).

There is existing work that extracts features based on product specification [7] and another one that discovers aspects and associated opinions [13] which can then be used to summarize reviewer's opinions for a single product, while we use coverage to automatically select snippets talking about important aspects. Or we can look at retrieving consensus opinion [5, 3] on product features, while our approach tries to show snippets covering the most important facts or opinions. We are using a coverage based objective to achieve diversity, because balance of presented aspects is important [6].

Other research has already been done in presenting summaries as pairs of items. For example, contrastive pairs [4, 9] aligns positive and negative opinions on the same aspect. Our approach does not produce a summary that contrasts sentiment, but we construct pairs that contrast products.

Another relevant topic to our work are approaches that summarize differences, such as comparative summaries constructed by using dominating sets [11] and scoring terms based on how likely they are to appear in the other collection [10]. Similar ideas can also be found in summarizing differences in multilingual news [12].

snippets for Apple iPad	snippets for Google Nexus
an ipod built in so you can listen to your favorite tunes via the music app or download new music via the itunes app	have to download specific apps to be able to download anything since safari doesnt handle downloads and you cant add music to the ipod app without first synching with itunes unless youre purchasing from itunes on the device itself
to sync my gmail contacts i had to set up my gmail account as a microsoft exchange account this is stupid why cant i just set it up as a gmail account and automatically sync my contacts without the extra hassle	setup was easy and it synced flawlessly with my gmail account automatically downloading apps
no microsd card slots hdmi or anything	miss from other tablets when using this one is the microsd expansion slots that so many android tablets have
millions of people use their computers for gaming and with the iphone and ipod touch having taken on a clear role as a gaming console that has been as revolutionary for mobile gaming as the wii was for livingroom gaming	you plan on gaming and storing music you may have a bit of trouble with running out of space
you care to pay for the extra space or connectivity is a matter of personal preference i opted for just the 16 gig wifi only model	if you think you might like some extra storage space then i suggest getting the new 32gb model that was recently released

Figure 2: Aligned snippet pairs selected by our method comparing two tablets.

If we compare related work with our approach, we are closer to summarizing differences except that snippets still have to be aligned on aspect. Compared to opinion focused approaches we do not distinguish between positive and negative, but focus on how is one product (perceivably) different (or similar) from the other. Although we implicitly try to cover all important aspects, we do not explicitly extract them and rely on submodular objective to achieve balanced coverage and avoid redundancy.

5. CONCLUSION

In this paper we presented an approach to selecting pairs of snippets from reviews in a way that creates a summarizing product comparison. Our scoring function strives to select aligned pairs (both snippets are about the same aspect) with good coverage of important aspects and low redundancy. The objective function is submodular and thus efficient to optimize with a constant factor approximation. Our experiments show that we outperform a naive baseline even with the uniform weights model. Using a supervised learning approach we can achieve generalization across different product pairs by using user feedback on the presented pairs.

6. ACKNOWLEDGEMENTS

This research was funded in part by NSF Awards IIS-0905467, IIS-1142251, and IIS-1247696.

7. REFERENCES

[1] P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan. Exploring sentiment summarization. In *AAAI Spring Symposium*, 2004.

[2] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, New York, NY, USA, 1998. ACM.

[3] J. Choi, D. Kim, S. Kim, J. Lee, S. Lim, S. Lee, and J. Kang. Consento: a new framework for opinion based entity search and summarization. In *CIKM*, pages 1935–1939, New York, NY, USA, 2012. ACM.

[4] H. D. Kim and C. Zhai. Generating comparative summaries of contradictory opinions in text. In *CIKM*, pages 385–394, New York, NY, USA, 2009. ACM.

[5] T. Lappas and D. Gunopulos. Efficient confident search in large review corpora. In *ECML PKDD*, pages 195–210, Berlin, Heidelberg, 2010. Springer-Verlag.

[6] M. Mahajan, P. Nguyen, and G. Zweig. Summarization of multiple user reviews in the restaurant domain. Technical Report MSR-TR-2007-126, Microsoft Research, 2007.

[7] X. Meng and H. Wang. Mining user reviews: from specification to summarization. In *ACL-IJCNLP Short Papers*, pages 177–180, Stroudsburg, PA, USA, 2009. ACL.

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

[9] M. J. Paul, C. Zhai, and R. Girju. Summarizing contrastive viewpoints in opinionated text. In *EMNLP*, pages 66–76, Stroudsburg, PA, USA, 2010. ACL.

[10] G. Raveendran and C. L. Clarke. Lightweight contrastive summarization for news comment mining. In *SIGIR*, pages 1103–1104, New York, NY, USA, 2012. ACM.

[11] C. Shen and T. Li. Multi-document summarization via the minimum dominating set. In *COLING*, pages 984–992, Stroudsburg, PA, USA, 2010. ACL.

[12] X. Wan, H. Jia, S. Huang, and J. Xiao. Summarizing the differences in multilingual news. In *SIGIR*, pages 735–744, New York, NY, USA, 2011. ACM.

[13] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *SIGKDD*, pages 783–792, New York, NY, USA, 2010. ACM.