GENERATING COMPARATIVE SUMMARIES OF CONTRADICTORY OPINIONS IN TEXT

BY

HYUN DUK KIM

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2009

Urbana, Illinois

Adviser:

Associate Professor ChengXiang Zhai

Abstract

This thesis presents a study of a novel summarization problem called *contrastive opinion summarization* (COS). Given two sets of positively and negatively opinionated sentences which are often the output of an existing opinion summarizer, COS aims to extract comparable sentences from each set of opinions and generate a comparative summary containing a set of contrastive sentence pairs. We formally formulate the problem as an optimization problem and propose two general methods for generating a comparative summary using the framework, both of which rely on measuring the content similarity and contrastive similarity of two sentences. We study several strategies to compute these two similarities. We also create a test data set for evaluating such a novel summarization problem. Experiment results on this test set show that the proposed methods are effective for generating comparative summaries of contradictory opinions. In addition, we implemented two demo systems which show the usefulness of the algorithm intuitively.

To my parents.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Prof. ChengXiang Zhai, for all his comments and helps for this research. He always supported me and gave me a lot of inspiration of research. Without his considerate guidance, this thesis could not be done. I also want to thank Jaehoon Lee, Dae Hoon Park, Hyejin Yang and Jang Hyeon Jo for human annotations of the experiment data, and other colleagues in Text Information Management group for useful comments.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

COS	Contrastive Opinion Summarization.
WO	Word Overlap.
SEM	Semantic Word Matching.
RF	Representativeness-First.
CF	Contrastiveness-First.

List of Symbols

$\pi(s_1, s_2)$ Content similarity function.

$\psi(u, v)$ Contrastive similarity function.

$r(S)$ Representativeness.

$c(S)$ Contrastiveness.

λ Parameter to control the relative importance of representativeness and contrastiveness with a larger λ indicating more emphasis on the representativeness.

$\omega(u, v)$ Term similarity function.

γ Paramter to control the strength of semantic matching.

Chapter 1

Introduction

With Web 2.0 technologies prevailing, people can now easily express opinions on various topics through platforms such as blog spaces, forums, and dedicated opinion websites. Since there is usually a large amount of opinionated text about a topic, users often find it challenging to efficiently digest all the opinions. The fact that opinionated text often contains both positive and negative opinions about a topic makes it even harder to accurately digest mixed opinions.

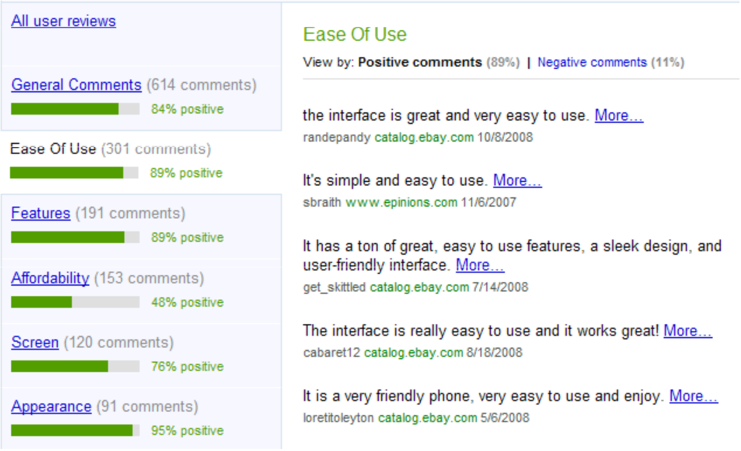
For example, some customers may say positive things about the battery life of iPhone, such as "the battery life [of iPhone] has been excellent," but others might say the opposite, such as "I can tell you that I was very disappointed with the 3G [iPhone] battery life."[1] Often such contradictory opinions are not caused by poor or wrong judgments of people, but due to the different context or perspective taken to make the judgments. For example, if a positive comment is 'the battery life is good when I rarely use button' and a negative comment is 'the battery life is bad when I use button a lot', the two comments are really made under different conditions. When there are many such contradictory opinions expressed about a topic, a user would need to understand what the major positive opinions are, what the major negative opinions are, why these people have different opinions, and how we should interpret these contradictory opinions.

Unfortunately although there has been much work on opinion summarization (see, e.g., [12, 17]), most existing work has gone only as far as separating positive and negative opinions about a topic.

For example, Figure 1 shows a part of a sample review summary generated using a state-of-the-art feature-based opinion summarization technique [7, 13]. In such an opinion summary, a user can see the general sentiment distribution for each product feature, and furthermore, as shown in the figure, a user can also see a list of positive comments about a specific feature (i.e., "ease of use"). Negative sentences are also available via another tab on the top. However, this summary cannot help a user to further digest the mixed opinions in the dimension of "ease of use". The user still has to read all the individual comments in both the positive and negative groups.

If opinions are dominated to one side, for example most people have positive opinions, general sum-

[1] These sentences are real examples found by the Products Live Search portal at http://search.live.com/products/.



All user reviews

General Comments (614 comments)
84% positive

Ease Of Use (301 comments)
89% positive

Features (191 comments)
89% positive

Affordability (153 comments)
48% positive

Screen (120 comments)
76% positive

Appearance (91 comments)
95% positive

Ease Of Use

View by: **Positive comments** (89%) | Negative comments (11%)

the interface is great and very easy to use. More...
randepandy catalog.ebay.com 10/8/2008

It's simple and easy to use. More...
sbraith www.epinions.com 11/6/2007

It has a ton of great, easy to use features, a sleek design, and user-friendly interface. More...
get_skittled catalog.ebay.com 7/14/2008

The interface is really easy to use and it works great! More...
cabaret12 catalog.ebay.com 8/18/2008

It is a very friendly phone, very easy to use and enjoy. More...
loretitoleyton catalog.ebay.com 5/6/2008

Figure 1.1: A sample state-of-the-art opinion summary (http://search.live.com/products/)

marization techniques can help to shrink the number of opinions to read. However, the techniques do not consider characteristics of opinionated data and cannot show important contradictions in mixed opinions effectively.

To help people digest such mixed opinions more efficiently, we propose to automatically generate a comparative summary of contradictory opinions. Specifically, given a set of positively opinionated sentences and a set of negatively opinionated sentences (which can be generated using existing techniques of opinion summarization), we would like to extract comparable sentences from each set of opinions and generate a comparative summary containing a set of contrastive sentence pairs. Each contrastive sentence pair consists of a sentence with positive opinions and a comparable sentence with negative opinions, thus enabling a user to understand contradictory opinions effectively. For example, if we can pair up two representative sentences with opposite opinions about the battery life of iPhone, it would help a user to understand possibly different conditions under which the specific polarity of opinions is expressed, and thus better understand why there are both positive and negative opinions about the battery life.

To the best of our knowledge, this summarization problem has not been addressed in the existing work, and we call it contrastive opinion summarization (COS). We formally formulate the COS problem as an optimization problem in which we attempt to find a list of contrastive sentence pairs that can both represent the two sets of opposite opinions well and offer interesting comparisons between positive and negative opinions about the same topical aspect (e.g., battery life). The objective function of the optimization framework encodes two criteria to be applied to choose sentence pairs. One is that a chosen sentence from the set of positive (negative) sentences should represent a major positive (negative) opinion, i.e., there should be many sentences similar to the chosen sentence. We call this criterion *representativeness*. The other is that the two paired sentences should be comparable. That is, they should have opposite opinions about a *common* topical aspect. We call this criterion *contrastiveness*.

Intuitively we need different similarity functions to measure representativeness and contrastiveness. While we can generally use an existing sentence similarity function to measure representativeness, we need a new similarity function to measure contrastiveness. We solve this problem by excluding sentimental words from both sentences and then applying a regular similarity function. We also explore how to leverage resources such as WordNet to accommodate matching of words that are semantically related but have different forms.

Exact solution to the optimization problem is generally intractable for realistic applications. We propose two general approximation methods to solve the problem. Both methods are greedy algorithms, corresponding roughly to first maximizing representativeness and then maximizing contrastiveness, or the opposite, i.e., first maximizing contrastiveness and then maximizing representativeness.

Because no existing data can be used directly to evaluate this new summarization task, we opted to create our own test set based on some publically available resources from the previous work [7, 8]. To test the generality of our methods, we further extended the test set by adding an additional case from a different domain.

Experiment results on this test set show that the proposed methods are effective for generating comparative summaries of contradictory opinions.

In addition to the experiments, we implemented two demo systems which can show the practical usefulness of the proposed method. The first demo system allows users to input their own data and applies the proposed algorithm to the user input. The second demo system is an opinion search engine using the proposed method. Users can query with search keyword and summarize opinions on the retrieved documents.

The contributions of this thesis are:

1. We propose and define a novel summarization problem (i.e., contrastive opinion summarization).

2. We propose an optimization framework to model and solve this problem.

3. We propose specific methods to solve the optimization problem and generate contrastive opinion summaries.

4. We create the first test set and propose measures for evaluating this novel summarization problem.

5. We run experiments to test the proposed methods and show that the proposed methods are effective.

6. We implemented demo systems and show the practical usefulness of the proposed method.

The rest of the thesis is organized as follows. In Chapter 2, we define the novel problem of contrastive opinion summarization. In Chapter 3, we formally model the problem with an optimization framework. In Chapter 4 and Chapter 5, we then present specific methods to refine and solve the optimization problem. We present our experiment design in Chapter 6 and results in Chapter 7, and show system implementation in Chapter 8. At last, we discuss related work in Chapter 9, and Chapter 10 concludes the work.

Chapter 2

Problem Definition

As discussed in the previous chapter, the current opinion summarization techniques can separate positive sentences from negative sentences about a topic (e.g., a product feature). We set up our problem as to take these sentences with different polarities as input and further generate a contrastive opinion summary to help users to digest the mixed opinions about the topic.

We thus will assume that we are given two sets of opinionated sentences about a topic, corresponding to positive and negative opinions about the topic, respectively. Our goal is to generate a list of sentence pairs with each pair containing a position sentence and a comparable negative sentence. Such a pair would allow a user to compare comparable positive and negative opinion and thus facilitate digestion of mixed opinions.

To formally define our problem, we first introduce a few basic concepts.

Definition 1 (Opinionated Sentence) *A sentence is an opinionated sentence if it expresses either a positive or a negative opinion. For convenience, we will simply call a positively (negatively) opinionated sentence a positive (negative) sentence.*

Definition 2 (Contrastive Sentence Pair) *A pair of opinionated sentences* (x, y) *is called a contrastive sentence pair if sentence* x *and sentence* y *are about the same topic aspect, but have opposite sentiment polarities.*

For example, x and y may both discuss the battery life of a laptop, but x says that that the battery life is long, while y says that it is short.

We may now define the novel problem of *contrastive opinion summarization.*

Definition 3 (Contrastive Opinion Summarization) *Let* $X = \{x_1, ..., x_n\}$ *be a set of positive sentences and* $Y = \{y_1, ..., y_m\}$ *be a set of negative sentences about a common topic* Q*, where* x_i *is a positive sentence and* y_i *is a negative sentence. The task of contrastive opinion summarization (COS) is to generate* k contrastive sentence pairs*:* $\{(u_i, v_i)\}$*,* $i = 1, ...k$*,* $u_i \in X$*,* $v_i \in Y$*, such that* $U = \{u_i\}_{i=1}^{k} \subset X$ *can represent the opinions in* X *well, and* $V = \{v_i\}_{i=1}^{k} \subset Y$ *can represent the opinions in* Y *well.*

Table 2.1: Illustration of a contrastive opinion summary

Contradictory Aspect	**Positive**	**Negative**
Contradictory 1	u_1	v_1
Contradictory 2	u_2	v_2
...	...	...
Contradictory k	u_k	v_k

Table 2.1 illustrates how we may display a contrastive opinion summary in a tabular format to facilitate digestion of contradictory opinions. Each pair $(u_i,\ v_i)$ summarizes a contradictory aspect. A user can use u_i and/or v_i as "entry points" to navigate into relevant discussion about each side of the opinions of the corresponding contradictory aspect.

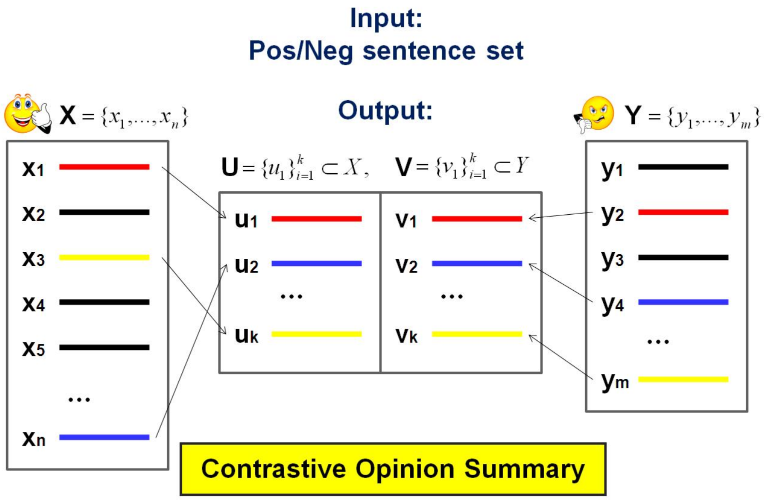


Figure 2.1: Overview of contrastive opinion summary

Figure 2.1 shows the overview of contrastive opinion summary. Among input sentences, alignable contrastive opinions are marked as different colors. Contrastive opinion summary extracts those sentences and aligned them to the matching opposite-oriented opinions.

Intuitively, to generate a good contrastive summary, we would need to match sentences in X with those in Y to discover potential candidate contrastive sentence pairs. At the same time, we also would like to assess which sentences can represent each polarity of opinions well. In the end, we would like to choose sentences from both X and Y that can not only form good contrastive pairs but also represent the corresponding complete set of opinions well. The problem is thus in nature an optimization problem involving multiple criteria. Below we will propose a formal optimization framework for solving COS, which would then use as a roadmap to derive several specific summarization algorithms.

Chapter 3

An Optimization Framework

In this chapter, we formally frame the contrastive opinion summarization problem as an optimization problem. Our optimization framework is based on two basic similarity measures defined on a pair of sentences. The first is to measure the content similarity of two sentences in the same group of opinions (i.e., either both are positive or both are negative). This similarity function allows us to assess which sentences are good representatives of each group. The second is to measure the contrastiveness of a positive sentence and a negative sentence. Since a good pair of contrastive sentences are generally also similar in content (but opposite in sentiment polarity), we also call this measure a *cross* group similarity measure. We formally define these two functions as follows.

Definition 4 (Content Similarity Function) *Given two opinionated sentences s_1 and s_2 with the same polarity, the content similarity function $\phi(s_1, s_2) \in [0, 1]$ measures the overall content similarity of s_1 and s_2.*

Definition 5 (Contrastive Similarity Function) *Given two opinionated sentences u and v with opposite polarities, the contrastive similarity function $\psi(u, v) \in [0, 1]$ measures the similarity of u and v excluding their difference in sentiment.*

Both ϕ and ψ are assumed to be symmetric, i.e., $\phi(s_1, s_2) = \phi(s_2, s_1)$, and $\psi(u, v) = \psi(v, u)$.

With these two functions, we can now define two additional functions that can measure the representativeness and contrastiveness of a contrastive opinion summary $S = \{(u_i, v_i)\}$ $(i = 1, ...k)$ of two sets of opinionated sentences X and Y.

Definition 6 (Representativeness) *The representativeness of a contrastive opinion summary S, denoted as $r(S)$, measures how well the summary S represents the opinions expressed by the sentences in both X and Y. It is defined as*

$$r(S) = \frac{1}{|X|} \sum_{x \in X} \max_{i \in [1,k]} \phi(x, u_i) + \frac{1}{|Y|} \sum_{y \in Y} \max_{i \in [1,k]} \phi(y, v_i).$$

Intuitively, if for every sentence $x \in X$, we have at least one u_i with high similarity to x, our summary would represent X well. Similar reasoning can be applied to the set Y. $r(S)$ is simply an aggregation over all the sentences in both X and Y.

Definition 7 (Contrastiveness) *The contrastiveness of a contrastive opinion summary S, denoted as $c(S)$, measures how well each u_i matches up with v_i in the summary. It is defined as the average contrastive similarity of the sentence pairs in S:*

$$c(S) = \frac{1}{k} \sum_{i=1}^{k} \psi(u_i, v_i)$$

A good contrastive opinion summary should intuitively have both high representativeness and high contrastiveness, thus we may cast the problem of contrastive opinion summarization as the following optimization problem:

$$\begin{aligned} S^* &= \arg\max_S \ (\lambda\, r(S) + (1-\lambda)\, c(S)) \\ &= \arg\max_S \left(\frac{\lambda}{|X|} \sum_{x \in X} \max_{i \in [1,k]} \phi(x, u_i) + \frac{\lambda}{|Y|} \sum_{y \in Y} \max_{i \in [1,k]} \phi(y, v_i) \right. \\ &\qquad \left. + \frac{1-\lambda}{k} \sum_{i=1}^{k} \psi(u_i, v_i) \right) \end{aligned}$$

where $\lambda \in (0, 1)$ is a parameter to control the relative importance of representativeness and contrastiveness with a larger λ indicating more emphasis on the representativeness.

With such an optimization framework, we see that in order to find an optimal contrastive opinion summary, we will need to solve three problems:

1. Define an appropriate content similarity function ϕ.
2. Define an appropriate contrastive similarity function ψ.
3. Solve the optimization problem efficiently.

In the next two chapters, we will discuss how we solve these problems.

Chapter 4

Similarity Functions

In this chapter, we discuss how we implement the two similarity functions ϕ and ψ. Our optimization framework allows us to flexibly implement them in any reasonable way as long as the two similarity functions can be normalized into the same range. This normalization is needed to ensure that the terms of these two functions in the objective function be comparable.

The content similarity function ϕ is meant to be a normal sentence similarity measure applied to two sentences in the same opinion group. In order to consider semantic matching of terms, we define $\phi(s_1, s_2)$ generally as:

$$\frac{\sum_{u \in s_1} \max_{v' \in s_2} \omega(u, v') + \sum_{v \in s_2} \max_{u' \in s_1} \omega(u', v)}{|s_1| + |s_2|}$$

where $\omega(u, v) \in [0, 1]$ is a term similarity function and $|s_1|$ and $|s_2|$ are the total counts of words in sentences s_1 and s_2, respectively.

The idea of this formula is that we would first match every word in each sentence against words in the other sentence to find its best matching score, then take a sum of all the matching scores, and finally normalize the sum by the total number of words in the two sentences. Since $\omega(u, v) \in [0, 1]$, clearly $\phi(s_1, s_2)$ is also in the range of $[0, 1]$.

Depending on how we define ω, we can obtain several different variations of this general similarity function. In this thesis, we will experiment with the following two natural variants:

1. **Word Overlap (WO):** $\omega_{WO}(u, v) = 1$ iff $u = v$, and $\omega_{WO}(u, v) = 0$ otherwise. In this case, ϕ would be essentially the Jaccard similarity function which only considers word overlap.

2. **Semantic Word Matching (SEM):** $\omega_{SEM}(u, v) = 1$ if $u = v$, and $\omega_{SEM}(u, v) = \gamma sim(u, v)$ otherwise, where γ is a parameter, and $sim(u, v)$ can be any semantic term similarity such as the value given by the WordNet:Similarity tool[1] [19], which we also use in our experiments. Clearly, if $\gamma = 0$, ω_{SEM} degenerates to ω_{WO}.

[1] http://www.d.umn.edu/ tpederse/similarity.html

The contrastive similarity function ψ is meant to measure how well two sentences with opposite opinions match up with each other. Intuitively, we would like the two sentences to overlap on all words except for those sentiment related words, where they are expected to differ. Thus, we define ψ in the same way as we define ϕ except that we now calculate the similarity *after* removing negation words and adjectives from both sentences. The rationale is that opinions are mainly expressed by adjectives and negation words. We also have two variations for ψ: WO and SEM.

Note that although in this thesis we only experiment with these simple similarity measures, our optimization framework would allow us to potentially use more sophisticated measures (e.g., [5, 21, 2, 15, 23]).

Chapter 5

Optimization Algorithms

A brute-force solution to the optimization problem defined in Chapter 3 would be to enumerate all the possible candidate summaries (i.e., k sentence pairs) and compute the objective function for each candidate summary. Since $|X| = n$ and $|Y| = m$, there are altogether $\binom{n}{k}\binom{m}{k}$ possible candidate summaries. Thus enumerating all of them is generally not feasible especially because computation of the value of the objective function for a candidate summary also involves additional iterations.

We now propose two heuristic strategies to find an approximate solution to the optimization problem. Our objective function contains two parts, corresponding to the representativeness ($r(S)$) and contrastiveness ($c(S)$) of a summary, respectively. Thus a greedy way to optimize the objective function can be to first find a subset of summaries that can score well with one of them, and then try to further select a good summary that can also score well for the other component. Depending on whether we would first optimize $r(S)$ or $c(S)$, we naturally have two heuristic strategies to generate a contrastive opinion summary, called representativeness-first and contrastiveness-first, respectively.

5.1 Representativeness-First Approximation

Figure 5.1 shows basic idea of representativeness-first approximation.

To optimize $r(S)$ means to select k sentences from each of X and Y that can best represent all the sentences in X and Y. Intuitively, we may achieve this goal by clustering the sentences in X and Y independently to generate k clusters for each, and then take the most representative sentence from each cluster. Specifically, let $\{U_1, ..., U_k\}$ be k clusters of sentences in X, and $\{V_1, ..., V_k\}$ be k clusters of sentences in Y. We may assume that $S = \{(u_i, v_i)\}$ where $u_i \in U_i$ and $v_i \in V_i$, for $i = 1, ..., k$. In general, given an implementation of the content similarity function ϕ, any clustering algorithm can be used. In our experiments, we used the hierarchical agglomerative clustering algorithm and stopped it when we have obtained precisely k clusters.

Now, we may reasonably assume that the similarity of a sentence to another sentence in a different cluster

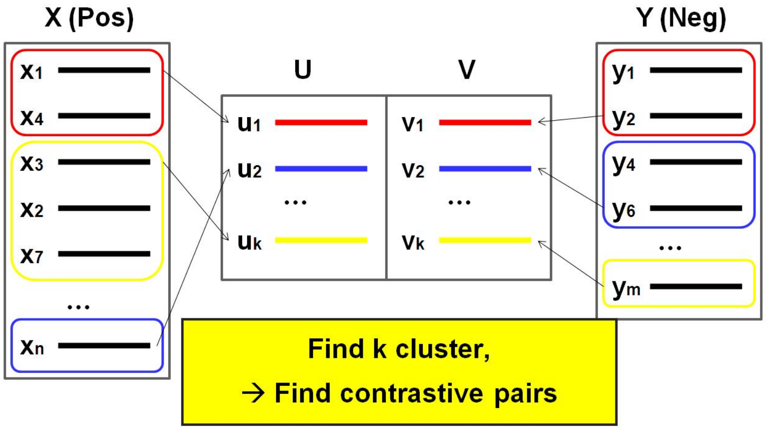


Figure 5.1: Overview of representativeness-first approximation

is always lower than its similarity to a sentence in its own cluster. It is not hard to prove that the summary S to maximize $r(S)$ is given by the centroid sentences in each cluster. That is, $S = \{\bar{u}_i, \bar{v}_i\}$, and

$$\begin{aligned}\bar{u}_i &= \arg\max_u \frac{1}{|X|}\sum_{x\in X}\phi(u,x)\\ \bar{v}_i &= \arg\max_v \frac{1}{|Y|}\sum_{y\in Y}\phi(v,y)\end{aligned}$$

Next we would like to keep $r(S)$ constant and optimize $c(S)$. Clearly $c(S)$ depends on how we index the clusters of X and Y, that is, how we order $\{U_1, ..., U_k\}$ and $\{V_1, ..., V_k\}$. In other words, it depends on how we align a cluster U_i with a cluster V_i. Intuitively we would like to align them so that the corresponding u_i and v_i would have the highest contrastiveness similarity, i.e., to maximize $\psi(u_i, v_i)$. Since k is generally small, we can find the exact optimal alignment without approximation.

One may notice that the weighting parameter λ did not matter in this strategy. Indeed, we have implicitly set $\lambda = \infty$ by first attempting to optimize $r(S)$ and then fix it to further optimize $c(S)$. To further improve over this, we may search in each aligned cluster pair to find a potentially better pair of sentences that can lead to a higher objective function value than the centroid pair defined above. If we do this, we will see that λ would affect our solution.

Specifically, let $\{U_1, ..., U_k\}$ and $\{V_1, ..., V_k\}$ be our optimal alignment of clusters. We may rewrite our objective function as $g(S) = \sum_{i=1}^{k} g_i(u_i, v_i)$

where $g_i(u_i, v_i)$ is given by

$$\lambda[\frac{1}{|X|}\sum_{x\in U_i}\phi(x,u_i)+\frac{1}{|Y|}\sum_{y\in V_i}\phi(y,v_i)]+\frac{1-\lambda}{k}\ \psi(u_i,v_i).$$

This objective function is now a sum over all the aligned cluster pairs, and we can now find the solution by optimizing each (u_i, v_i) pair independently. Formally,

$$(u_i^*, v_i^*) = \arg\max_{u_i\in U_i, v_i\in V_i} g_i(u_i, v_i).$$

Clearly $g_i(\bar{u}_i, \bar{v}_i)$ is not necessarily optimal, so we would like to search in U_i and V_i for a truly optimal (u_i^*, v_i^*). The brute force search has a complexity of $O(|U_i||V_i|)$, but we do not have to try every pair of sentences. Instead, we only need to try those pairs with a higher contrastiveness score than our centroid pair $(\bar{u}_i, \bar{v}_i)$ because if a pair has a lower contrastiveness score than the centroid pair, it would be impossible for it to have a higher g_i value. Thus computationally, we can simply sort all the pairs in each pair of clusters in the descending order of contrastiveness scores and then go down the list to compute its g_i value, until we hit the centroid pair. The pair that gives the highest g_i would be taken as (u_i^*, v_i^*). We do this for each cluster pair to obtain the optimal summary $S^* = \{(u_i^*, v_i^*)\}$.

5.2 Contrastiveness-First Approximation

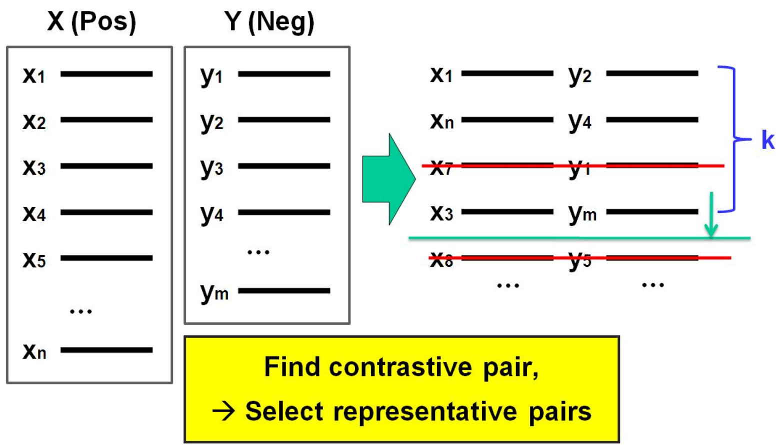


Figure 5.2: Overview of contrastiveness-first approximation

Figure 5.2 shows basic idea of contrastiveness-first approximation.

In this strategy, we first compute $\psi(u, v)$ for all $u \in X$ and $v \in Y$. We then sort these pairs and gradually add a sentence pair to our summary starting with the pair with the highest contrastive similarity score. If we just take the top k pairs, we would find a solution corresponding to setting $\lambda = 0$, i.e., solely based on contrastiveness and ignoring representativeness completely. Thus to improve our approximation, we would like to sacrifice some amount of contrastiveness score and gain more on the representativeness score.

A greedy algorithm to achieve this is as follows. First, we would take the pair with the highest value of ψ as a pair in our summary. Given that we have already chosen $i-1$ pairs $S_{i-1} = \{(u_j, v_j)\}_{j=1}^{i-1}$, we would like to choose the next pair (u_i, v_i) to add most to our objective function, which further means to maximize the increase of both the contrastiveness and the representativeness. Given a candidate pair (u_i, v_i), and let S_i be the augmented summary of S_{i-1} by adding this new pair. We want to choose (u_i, v_i) to maximize the following objective function :

$$\begin{aligned}(u_i^*, v_i^*) &= \arg\max_{u_i, v_i} \lambda r(S_i) + (1-\lambda) c(S_i) \\ &= \arg\max_{u_i, v_i} \lambda(\frac{1}{|X|} \sum_{x \in X_{u_i}} \phi(x, u_i) + \frac{1}{|Y|} \sum_{y \in Y_{v_i}} \phi(y, v_i)) \\ &\quad + \frac{1-\lambda}{k} \psi(u_i, v_i)\end{aligned}$$

where X_{u_i} (Y_{v_i}) is the set of sentences in X (Y) that are closer to u_i (v_i) than to any previously chosen u_j (v_j), $j = 1, ..., i-1$. That is,

$$X_{u_i} = \{x \in X | \phi(x, u_i) > \phi(x, u_j) \forall j = 1, ..., i-1\}$$

$$Y_{v_i} = \{y \in Y | \phi(y, v_i) > \phi(y, v_j) \forall j = 1, ..., i-1\}.$$

Thus in our greedy algorithm, after choosing the first pair (u_1, v_1), we would iteratively choose (u_i, v_i) to maximize the "gain function", $g(u_i, v_i)$ given by

$$\lambda(\frac{1}{|X|} \sum_{x \in X_{u_i}} \phi(x, u_i) + \frac{1}{|Y|} \sum_{y \in Y_{v_i}} \phi(y, v_i)) + \frac{1-\lambda}{k} \psi(u_i, v_i).$$

The algorithm stops after having chosen k pairs.

Note that X_u and Y_v are relatively easy to compute if we remember the best ϕ values of all sentences given by the $i-1$ already chosen sentence pairs at each step because we only need to compare the remembered value with the new value of ϕ given by u_i or v_i.

In general, we would need to consider all the remaining sentence pairs in each step. However, we can further improve efficiency by only considering the sentence pairs whose contrastiveness scores are sufficiently high (e.g., above a threshold).

Chapter 6

Experiment Design

6.1 Data Set

There is no existing data set for evaluating our new summarization task. We thus opt to create our own. Since a main assumption made in our problem definition is that we may separate positive and negative opinions about a topic using existing opinion summarization methods, a natural strategy to create a test set for evaluating COS would be to leverage such separated opinion data set generated by previous work. We thus obtained 14 tagged product review data sets from the previous work [7, 8][1], and have two human assessors to identify representative contrastive sentence pairs from these data. The 14 tagged review data sets contain reviews from Amazon[2]. All the sentences in these data sets have already been manually tagged with product features as well as sentiment polarities (i.e., positive or negative).

To show our methods can help users further understand opinions at a finer granularity level than the feature-level, we divided a product review into product-feature reviews based on the feature tags. Also, to make contrastive opinion summarization interesting, we chose reviews which are not extremely dominated by only positive (or negative) opinions using a threshold. Our assumption is that in those cases where reviews are dominantly of one polarity of opinions (e.g., dominantly positive), a regular summary would suffice, and we do not need to apply contrastive opinion summarization techniques. Because our data set is for evaluating the effectiveness of COS, we discarded extremely dominated sets. Based on these criteria, we obtained 12 review sets on several products and features.

To test the generality of our methods, we also prepared another non-product-review data set. We used the Yahoo! search engine to retrieve sentences about Aspartame, which is an artificial sweetener, and there are disputes about its safety. The constructed data set contains 50 positive and 50 negative matching sentences selected from the search results of the queries `‘aspartame is safe’` and `‘aspartame is dangerous’`, respectively.

Table 6.1 shows the data set list. For each data set, ID is assigned for convenience.

[1] http://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

[2] http://www.amazon.com

Table 6.1: Data set

ID	Product Name	Feature	# of pos sen	# of neg sen
1	Apex AD2600 Progressive-scan DVD player	player	44	56
2	MicroMP3	battery-life	9	7
3	MicroMP3	design	8	6
4	MicroMP3	headphones	7	6
5	MicroMP3	software	7	9
6	Nokia 6600	battery-life	7	8
7	Creative Labs Nomad Jukebox Zen Xtra 40GB	navigation	9	8
8	Creative Labs Nomad Jukebox Zen Xtra 40GB	software	37	41
9	Creative Labs Nomad Jukebox Zen Xtra 40GB	size	15	11
10	Creative Labs Nomad Jukebox Zen Xtra 40GB	weight	7	7
11	Creative Labs Nomad Jukebox Zen Xtra 40GB	transfer	9	7
12	Hitachi router	adjustment	7	6
13	aspartame	safety	50	50

For each test case, the two assessors were asked to cluster given sentences of each polarity of sentiment and align the contrastive clusters. Among the judgments made by human evaluators, clusters that cannot be aligned are discarded. Figure 6.1 shows how human assessor annotation are generated.

To assess the agreement of the two assessors, we compute their clustering agreement and pairing agreement, which are 0.76 and 0.47, respectively. The clustering agreement is the percentage of agreed decisions on putting a pair of sentences of the same polarity into the same cluster or not by the two annotators. In the original sentence sets, we can make same-polarity-sentence pairs, (u_i, u_j) where both u_i and u_j are positive (negative). For all the possible pairs, check if they are in the same cluster or not based on the two evaluators' judgments. If the judgments are same, it is an agreement. Then, clustering agreement is

$$\frac{\#\ of\ clustering\ agreement}{\#\ of\ all\ possible\ pairs\ of\ same\ polarity\ sentences}$$

The pairing agreement is computed using the Jaccard Index. First, we generate all the ideal contrastive sentence pairs from the evaluators' judgements on clustering and pairing. For example, if two clusters, $\{u_i, u_j\}$ and $\{v_l, v_m\}$, are paired, (u_i, v_l), (u_i, v_m), (u_j, v_l), and (u_j, v_m) would be generated as ideal pairs. Let A and B be the two sets of ideal pairs from two assessors, respectively, the Jaccard Index was calculated by the following formula

$$JaccardIndex(A, B) = \frac{A\ and\ B}{A\ or\ B}$$

In our experiments, we use each assessor's judgments separately for evaluation and then take the average of the two performance numbers. The constructed data sets are publically available[3].

[3] http://timan.cs.uiuc.edu/data/cos/

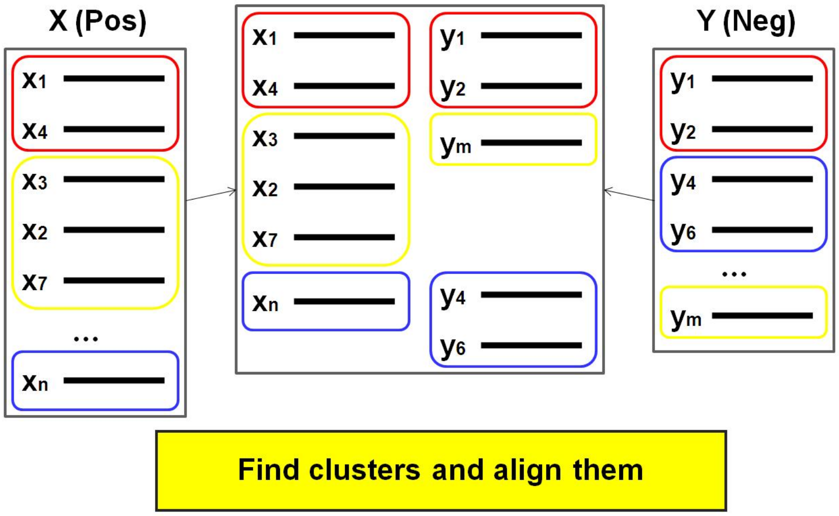


Figure 6.1: Overview of human assessor annotation

6.2 Measures

We evaluate our contrastive opinion summary with the following two measures based on the aligned cluster data set:

Precision: The precision of a summary with k contrastive sentence pairs is the percentage of the k pairs that are agreed by a human annotator. If a retrieved pair exists in an evaluator's paired-cluster set, we assume that the pair is agreed by the annotator (i.e., "relevant"). Thus precision is basically the number of such agreed pairs divided by k. Intuitively, precision tells us how contrastive the sentence pairs of our summary are.

Aspect coverage: The aspect coverage of a summary is the percentage of human-aligned clusters covered in the summary. If a pair of sentences appears in a human-aligned pair of clusters, we would assume that the aligned cluster is covered. Intuitively, aspect coverage measures the representativeness of a summary.

The number k of a target summary was set heuristically by the following formula; $k = 1+\log_2(|X|+|Y|)$. The intuition is that we will have a larger k if we have more sentences to summarize, but the growth will "saturate" as the number of sentences becomes very large.

6.3 Questions to Answer

We design our experiments to answer the following questions: First, between representative-first approximation(RF) and contrastive-first approximation(CF), which optimization algorithm performs better? We can answer this question by comparing the performance of these two different approximations for various values of λ. Second, both ϕ and ψ can use semantic matching of words. Can semantic matching of words help improve performance on top of simple exact matching of words? We can answer this question by comparing the performance of methods using different semantic coefficient γ. Third, we have hypothesized that it would be beneficial to exclude sentimental words when computing the contrastive similarity. Is this heuristic effective? We can answer this question by comparing performance of excluding such words with that of not excluding them (i.e., using all the words).

Chapter 7

Experiment Results

7.1 Sample Results

Table 7.1: Sample contrastive sentence pairs

No	Positive	Negative
1	oh ... and file transfers are fast & easy .	you need the software to actually transfer files
2	i noticed that the micro adjustment knob and collet are well made and work well too.	the adjustment knob seemed ok, but when lowering the router, i have to practically pull it down while turning the knob.
3	the navigation is nice enough , but scrolling and searching through thousands of , tracks hundreds of albums or artists , or even dozens of genres is not conducive to save driving .	difficult navigation - i wo n't necessarily say " difficult ," but i do n't enjoy the scrollwheel to navigate .
4	i imagine if i left my player untouched (no backlight) it could play for considerably more than 12 hours at a low volume level.	there are 2 things that need fixing first is the battery life. it will run for 6 hrs without problems with medium usage of the buttons.

We first show some sample contrastive sentence pairs generated in our experiments in Table 7.1. Intuitively, these pairs are quite informative and can clearly help a user better understand the mixed opinions in different aspects. The first and second pairs show that different polarities of opinions are made from different perspectives. For example, from the first pair, a user would know that file transfer is fast, but you'll need transfer software. Similarly, the second pair shows that adjustment knob generally works well, but it is inconvenient when lowering the router. In the third pair, although the two sentences are classified as positive and negative, respectively, the difference is rather small, indicating that there is probably not that much disagreement here. In the fourth pair, we can learn even more details about the product. This example shows the battery life can vary depending on usage patterns even with specific number of hours it can last; users can decide whether to buy this product based on their own usage style.

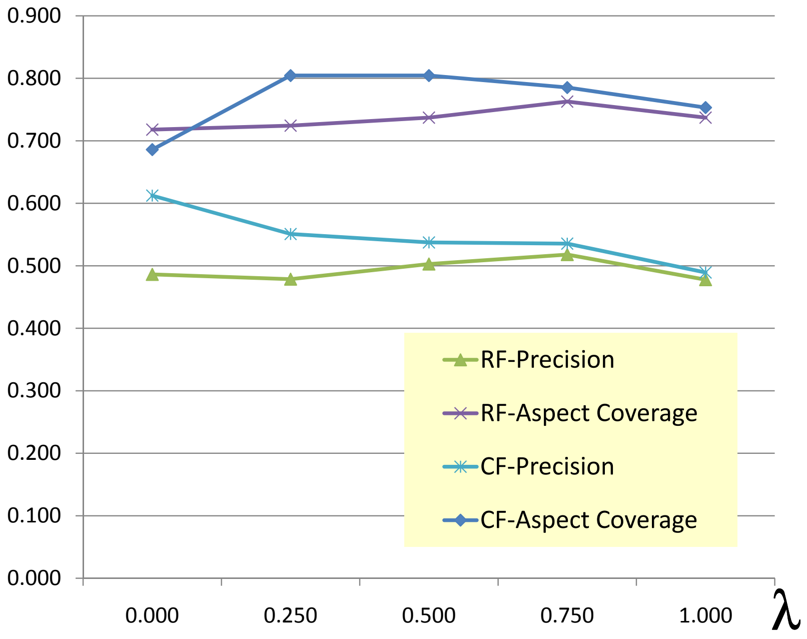


Figure 7.1: Comparison of RF and CF

7.2 Representativeness-First vs. Contrastiveness-First

Next, we use the basic similarity measure (ω_{WO}) to compare the two approximation strategies, i.e., representativeness-first (RF) and contrastiveness-first (CF) in Figure 7.1. For both methods, aspect coverage is higher than precision, indicating that it is easier to achieve representativeness than contrastiveness. In general, we see that CF outperforms RF for almost all values of λ, indicating that it is more important to optimize contrastiveness-first to ensure that we obtain the best contrastive alignments of sentences. We also see that the performance is sensitive to the setting of λ, the relative emphasis on the representativeness and contrastiveness of the summary. In order to examine other variables in our methods, in the following experiments, we set λ to a reasonable value of 0.5, which intuitively means that we put equal weights on the two criteria.

7.3 Semantic Term Matching

We now look into the effectiveness of semantic term matching. Since ω_{WO} is a special case of ω_{SEM} when $\gamma = 0$, we can see whether semantic matching helps by varying the value of γ. We show the results of using semantic term matching for content similarity and contrastive similarity respectively with two separate plots in Figure 7.2. In general, semantic term matching does not seem to help. Indeed, as we increase the value of γ, the performance tends to drop.

There can be several possible reasons of the result. Using semantic matching can incorporate noise because semantic matching may assign the non-zero score between not tightly related words. Rather small size of data can be other reason that the effect of semantic matching does not show clearly. In addition, the score change from semantic matching may not be big enough to affect sentence selection.

We also see that the content similarity function is more sensitive to semantic matching than the contrastive similarity function. This may be because in the latter case, sentimental words are removed, so the overall influence of semantic matching would be reduced.

7.4 Contrastive Similarity Heuristic

We hypothesized that by removing sentimental words in computing the contrastive similarity function we can improve matching accuracy. So finally, in order to test this hypothesis, we compare the results of using this heuristic with those of not using it (i.e., computing the contrastive similarity in the same way as computing the content similarity) in Table 7.2. We see that if we keep all these sentimental words, the performance is consistently worse, indicating that the heuristic of removing sentimental words is effective.

Table 7.2: Effectiveness of removing sentimental words in computing contrastive similarity.

	Precision		Aspect Coverage	
Opt. Method	RF	CF	RF	CF
WO	0.503	0.537	0.737	0.804
WO + all words	0.484	0.531	0.737	0.798
SEM	0.500	0.540	0.763	0.763
SEM + all words	0.470	0.507	0.718	0.686

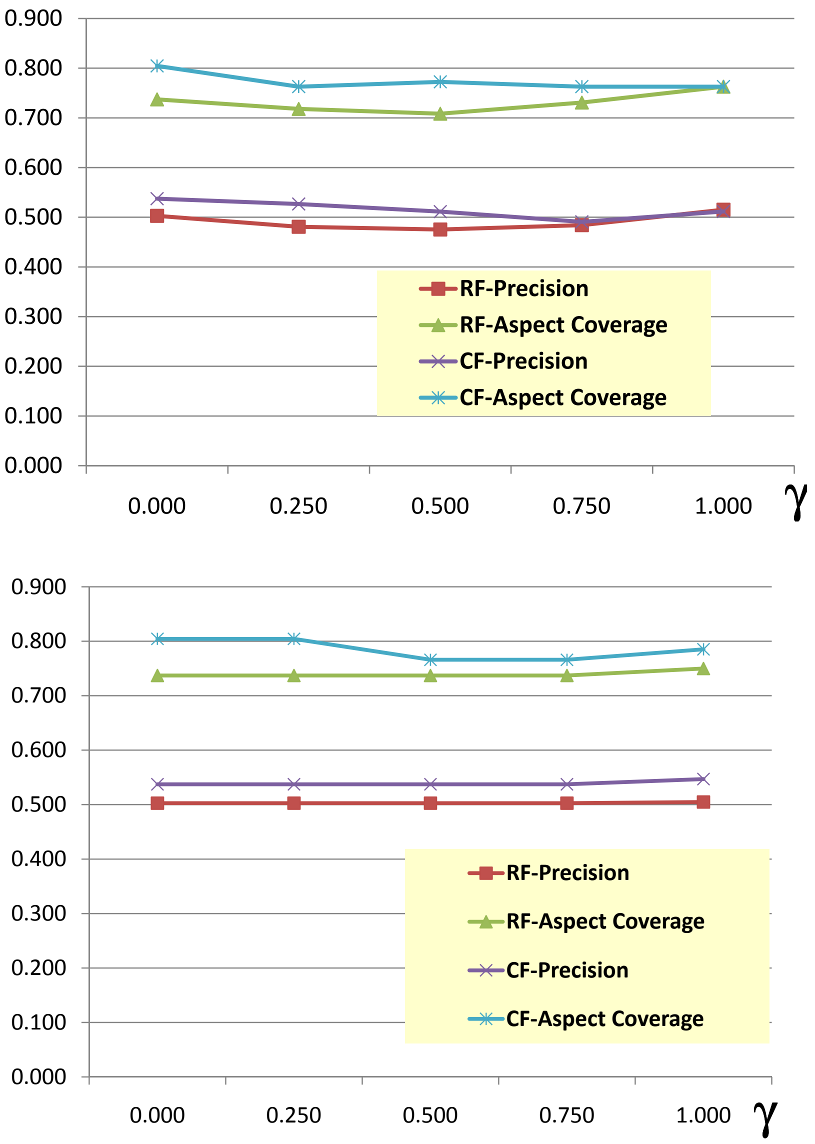


Figure 7.2: Effectiveness of semantic term matching for content similarity (top) and contrastive similarity (bottom)

Chapter 8

System Implementation

To increase the impact of the suggested algorithm and show more intuitive usefulness, we implemented two demo systems using the proposed algorithm. They can help people understand people the usefulness of the algorithm intuitively. In addition, the feedback information like user query and click through logs can be precious information for the further improvement of the current system and potentially can be used for the future research. Both demo systems are available at http://timan.cs.uiuc.edu/proj/cos/.

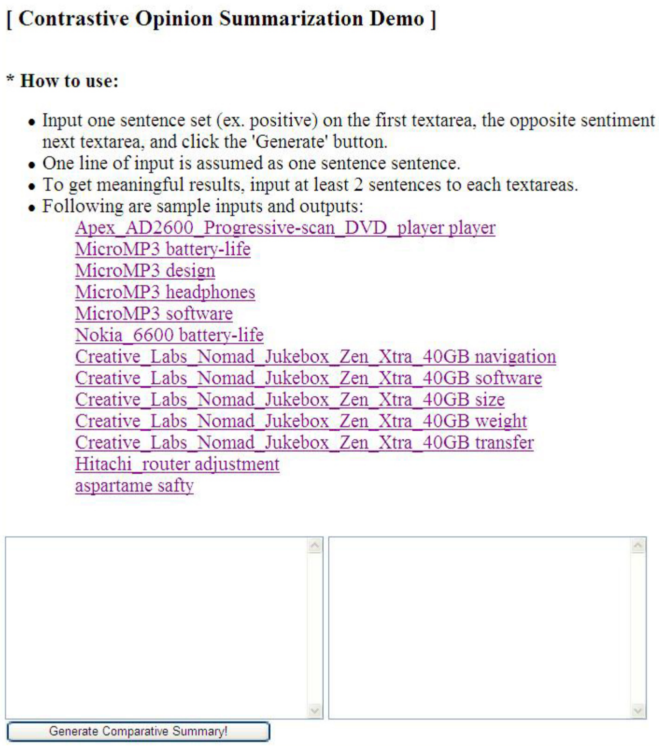
8.1 Contrastive Opinion Summary Generation System

8.1.1 System Overview

Figure 8.1 shows the input screen of the demo system. Users can try to generate a contradictory opinion summary by inserting their own sentence sets on the bottom text areas. According to experiment results, we used CF approximation with no semantic matching and $\lambda = 0.5$. There are also links to sample results from the data sets used in the experiments. Following is the usage of the demo system: input one sentence set (e.g., positive) on the first text area, the opposite sentiment sentence set (e.g., negative) on the next text area, and click the 'Generate' button. One line of input is assumed as one sentence.

8.1.2 System Architecture

Figure 8.2 shows the overall architecture of the system. The demo system is mainly implemented by HTML, php and Python. The main HTML page takes user inputs and passes the information to the next php page. The php page executes Python module on the server and shows the result returned from the module. For the table formatting, we use Google visualization API [1].



[Contrastive Opinion Summarization Demo]

*** How to use:**

- Input one sentence set (ex. positive) on the first textarea, the opposite sentiment sentence set (ex. negative) on the next textarea, and click the 'Generate' button.
- One line of input is assumed as one sentence sentence.
- To get meaningful results, input at least 2 sentences to each textareas.
- Following are sample inputs and outputs:
  Apex_AD2600_Progressive-scan_DVD_player player
  MicroMP3 battery-life
  MicroMP3 design
  MicroMP3 headphones
  MicroMP3 software
  Nokia_6600 battery-life
  Creative_Labs_Nomad_Jukebox_Zen_Xtra_40GB navigation
  Creative_Labs_Nomad_Jukebox_Zen_Xtra_40GB software
  Creative_Labs_Nomad_Jukebox_Zen_Xtra_40GB size
  Creative_Labs_Nomad_Jukebox_Zen_Xtra_40GB weight
  Creative_Labs_Nomad_Jukebox_Zen_Xtra_40GB transfer
  Hitachi_router adjustment
  aspartame safty

Figure 8.1: Contrastive opinion summarization generation system. Input page screen shot

8.2 Ooops!, Open Opinion Search System

To further demonstrate the usefulness of the proposed algorithm for summarizing search results, we developed Ooops!, which is an open opinion search engine with opinion summarization functions. The main screen of Ooops! (Figure 8.3) is similar to general search engines. If users type in search keywords into the text area and click search button, Ooops! retrieves relevant opinion documents in the search result page (Figure 8.4).

On the search result page, there is an advanced function menu below the search box. These functions further analyze the opinions of retrieved documents and show results.

Opinion filter function generates filtered opinion sentences of documents in the current search result page (Figure 8.5). Users can see all the opinion sentences of the each document, positive opinion sentences only, negative opinion sentences only, mixed opinion sentences only, or positive and negative sentences only (excluding mixed opinion sentences) by clicking each option. Each sentence is marked with the opinion orientation; (+) positive, (-) negative and (+-) mixed opinion.

Graph opinion summary function generates a visual opinion summary of the retrieved documents in the current search result page (Figure 8.6). The function adds a pie graph and table statistics between the

[1] http://code.google.com/intl/en/apis/visualization/

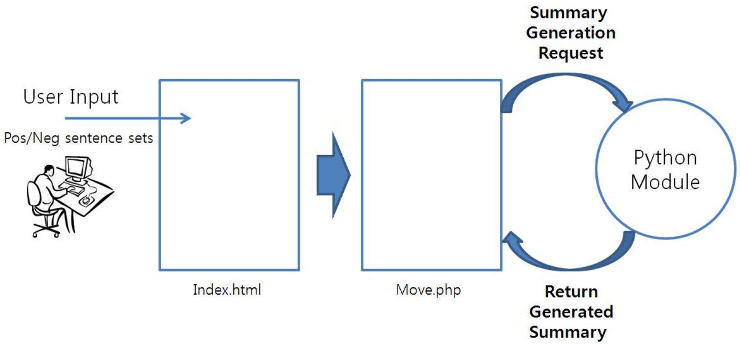
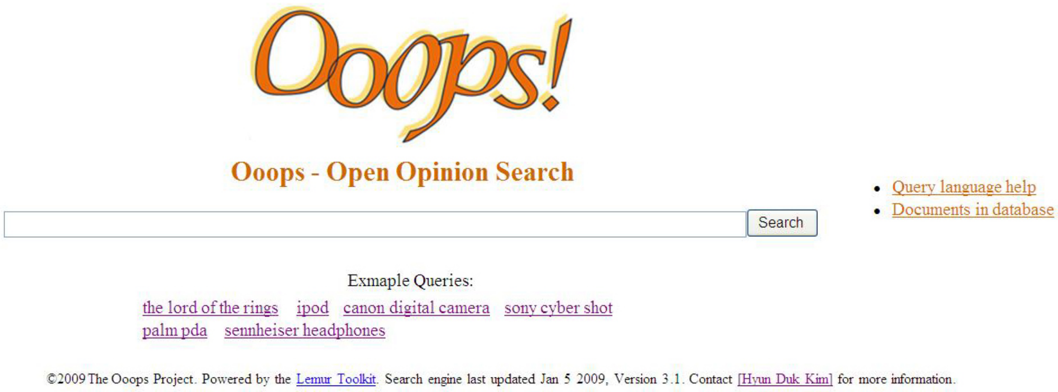


Figure 8.2: Contrastive opinion summarization generation system architecture



Ooops - Open Opinion Search

- Query language help
- Documents in database

Exmaple Queries:

the lord of the rings ipod canon digital camera sony cyber shot
palm pda sennheiser headphones

©2009 The Ooops Project. Powered by the Lemur Toolkit. Search engine last updated Jan 5 2009, Version 3.1. Contact [Hyun Duk Kim] for more information.

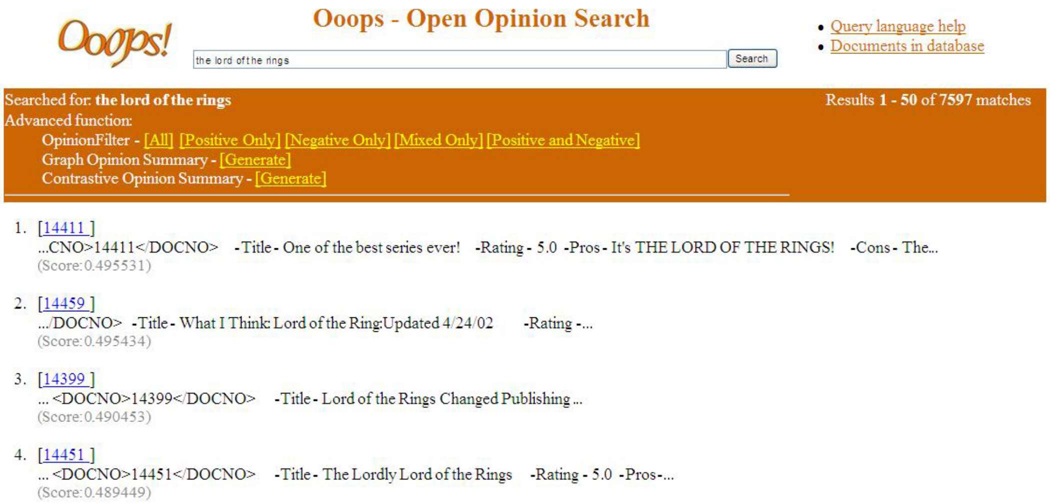
Figure 8.3: Ooops! main page screen shot

function menu and search results. They show the portion of each positive, negative and mixed opinions in retrieved documents in the current search result page. Below the two visual presentations, search results also show opinion sentences as we use the opinion filter function with the 'All' opinion option.

Contrastive opinion summary function generates a contrastive opinion summary of the retrieved document in the current search result page (Figure 8.7). The function adds a table containing contrastive opinion sentence pairs between the function menu and search results. Below the contrastive opinion summary table, search results also show positive and negative opinion sentences as we use opinion filter function with the 'Positive and Negative' opinion option. For the graph and table formatting, we use Google visualization API[2].

For the contrastive opinion summary generation, we use the contrastive opinion summarization algorithm introduced in this thesis. According to experiment results of the algorithm, we use CF approximation with no semantic matching and $\lambda = 0.5$ which showed the best performance.

[2] http://code.google.com/intl/en/apis/visualization/



Ooops - Open Opinion Search

the lord of the rings | Search

- Query language help
- Documents in database

Searched for: **the lord of the rings** Results **1 - 50** of **7597** matches

Advanced function:

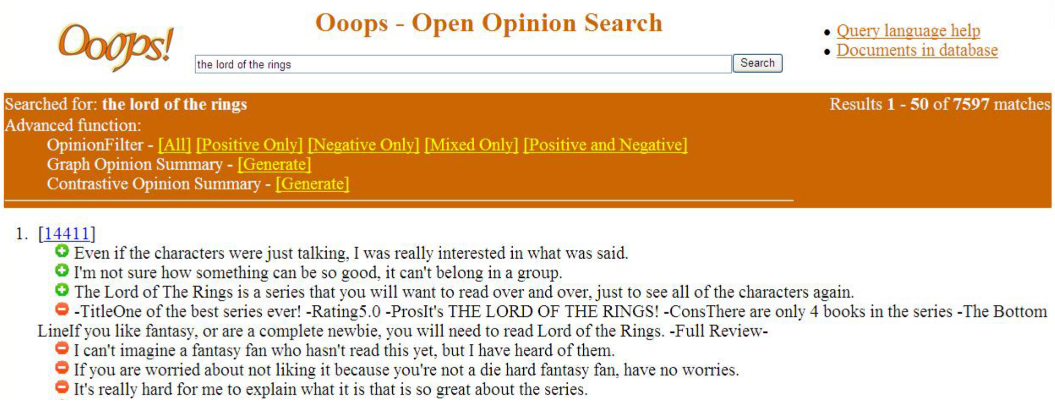
OpinionFilter - [All] [Positive Only] [Negative Only] [Mixed Only] [Positive and Negative]

Graph Opinion Summary - [Generate]

Contrastive Opinion Summary - [Generate]

1. [14411]
...CNO>14411</DOCNO> -Title - One of the best series ever! -Rating - 5.0 -Pros - It's THE LORD OF THE RINGS! -Cons - The...
(Score:0.495531)
2. [14459]
.../DOCNO> -Title - What I Think: Lord of the Ring:Updated 4/24/02 -Rating -...
(Score:0.495434)
3. [14399]
... <DOCNO>14399</DOCNO> -Title - Lord of the Rings Changed Publishing ...
(Score:0.490453)
4. [14451]
... <DOCNO>14451</DOCNO> -Title - The Lordly Lord of the Rings -Rating - 5.0 -Pros-...
(Score:0.489449)

Figure 8.4: Ooops! normal search result screen shot



Ooops - Open Opinion Search

the lord of the rings | Search

- Query language help
- Documents in database

Searched for: **the lord of the rings** Results **1 - 50** of **7597** matches

Advanced function:

OpinionFilter - [All] [Positive Only] [Negative Only] [Mixed Only] [Positive and Negative]

Graph Opinion Summary - [Generate]

Contrastive Opinion Summary - [Generate]

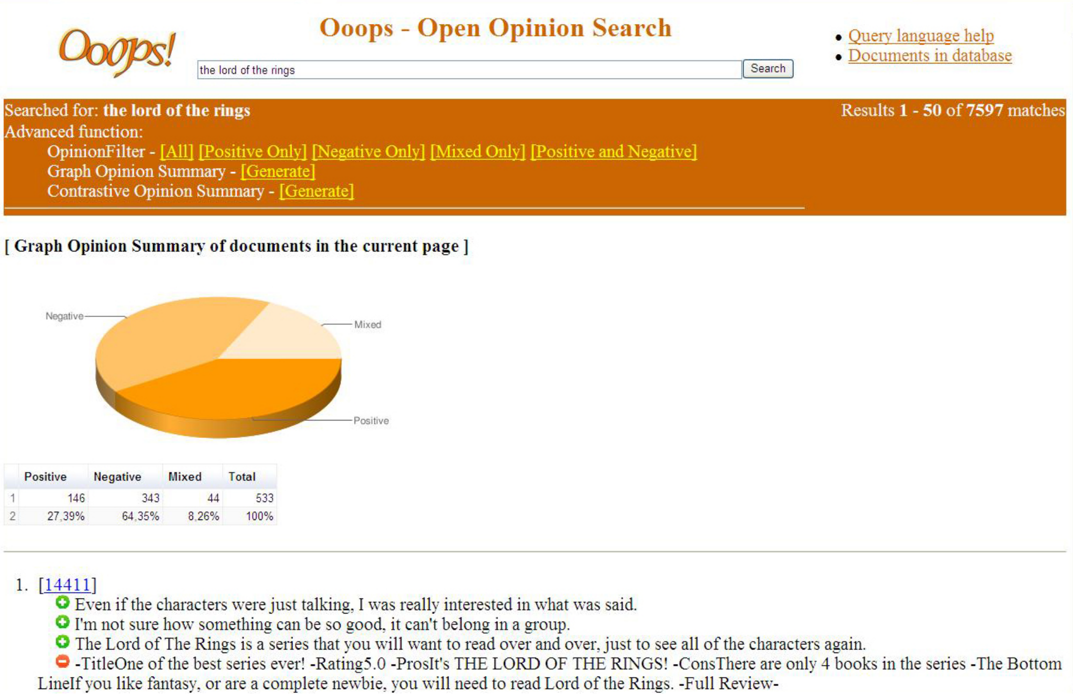
1. [14411]
⊕ Even if the characters were just talking, I was really interested in what was said.
⊕ I'm not sure how something can be so good, it can't belong in a group.
⊕ The Lord of The Rings is a series that you will want to read over and over, just to see all of the characters again.
⊖ -TitleOne of the best series ever! -Rating5.0 -ProsIt's THE LORD OF THE RINGS! -ConsThere are only 4 books in the series -The Bottom LineIf you like fantasy, or are a complete newbie, you will need to read Lord of the Rings. -Full Review-
⊖ I can't imagine a fantasy fan who hasn't read this yet, but I have heard of them.
⊖ If you are worried about not liking it because you're not a die hard fantasy fan, have no worries.
⊖ It's really hard for me to explain what it is that is so great about the series.

Figure 8.5: Ooops! opinion filtering screen shot

Because generating contrastive opinion summary needs much computation, to increase the responsiveness of the system, we use up to top 10 positive and negative opinions on the current search result page. Administrators can adjust this limit depending on purpose of the summary. Although limiting the number of opinions used for generating summary may decrease the completeness of the summary, it cannot only provide results faster but also generate more topic related summary because it uses opinions from only top ranked documents.

8.2.1 System Architecture

For the initial system, we collected about 13,000 reviews of 6 different product groups including book, mixer, mp3, digital camera, headphone and PDA from Epinions.com. All the reviews are cleaned, so only essential



Searched for: **the lord of the rings**

Results **1 - 50** of **7597** matches

Advanced function:

OpinionFilter - [All] [Positive Only] [Negative Only] [Mixed Only] [Positive and Negative]

Graph Opinion Summary - [Generate]

Contrastive Opinion Summary - [Generate]

[Graph Opinion Summary of documents in the current page]

	Positive	Negative	Mixed	Total
1	146	343	44	533
2	27.39%	64.35%	8.26%	100%

1. [14411]
   - ⊕ Even if the characters were just talking, I was really interested in what was said.
   - ⊕ I'm not sure how something can be so good, it can't belong in a group.
   - ⊕ The Lord of The Rings is a series that you will want to read over and over, just to see all of the characters again.
   - ⊖ -TitleOne of the best series ever! -Rating5.0 -ProsIt's THE LORD OF THE RINGS! -ConsThere are only 4 books in the series -The Bottom LineIf you like fantasy, or are a complete newbie, you will need to read Lord of the Rings. -Full Review-

Figure 8.6: Ooops! graph opinion summary generation screen shot

review texts such as title, pros, cons and full review text are remained. The coverage of the system can be extended with additional data collecting.

Figure 8.8 shows the overall architecture of the system. The main search system is implemented by using CGI interface of Lemur tool kit [3] (ver. 4.8). Collected and cleaned documents are indexed with the Lemur BuildDocMgr tool, and we use the default lemur retrieval model with CGI interface.

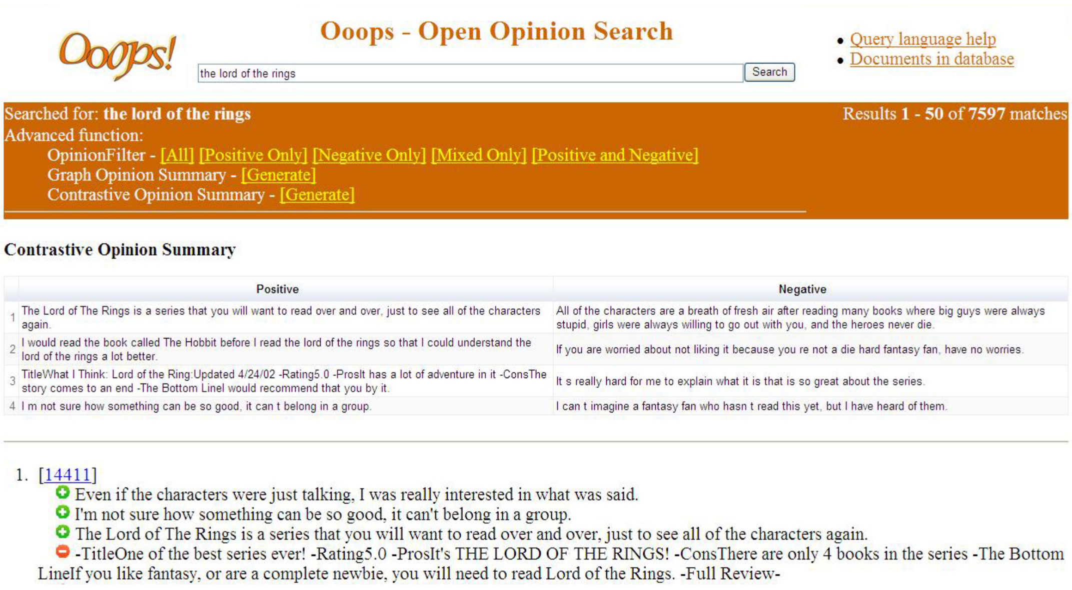
For the advanced opinion analysis function, each document is preprocessed by an opinion module. For the opinion classification, we use polarity classifier [25] of opinion finder tool [4] (ver. 1.5). After splitting sentences of original documents with a sentence segmentation tool [5], we check the opinion polarity of each sentence. Opinion finder marks sentiment words in the input sentence with their polarity. Therefore, there can be several polarities marked in one sentence. If all marked polarities are positive (or negative), we marked the sentence as positive (or negative). If there are both positive and negative polarities in one sentence, we assume the sentences as having 'mixed' sentiments. We assume the sentence having no polarity marks as 'neutral'. In summary, one sentence can have one of four polarities; positive, negative, mixed or neutral.

When we execute opinion filtering or opinion summary generation, we aggregate the opinions of each

[3] http://www.lemurproject.org/

[4] http://www.cs.pitt.edu/mpqa/opinionfinderrelease/

[5] http://l2r.cs.uiuc.edu/ cogcomp/atool.php?tkey=SS



Ooops - Open Opinion Search

- Query language help
- Documents in database

the lord of the rings [Search]

Searched for: **the lord of the rings** Results **1 - 50** of **7597** matches

Advanced function:

OpinionFilter - [All] [Positive Only] [Negative Only] [Mixed Only] [Positive and Negative]

Graph Opinion Summary - [Generate]

Contrastive Opinion Summary - [Generate]

Contrastive Opinion Summary

	Positive	Negative
1	The Lord of The Rings is a series that you will want to read over and over, just to see all of the characters again.	All of the characters are a breath of fresh air after reading many books where big guys were always stupid, girls were always willing to go out with you, and the heroes never die.
2	I would read the book called The Hobbit before I read the lord of the rings so that I could understand the lord of the rings a lot better.	If you are worried about not liking it because you re not a die hard fantasy fan, have no worries.
3	TitleWhat I Think: Lord of the Ring:Updated 4/24/02 -Rating5.0 -ProsIt has a lot of adventure in it -ConsThe story comes to an end -The Bottom LineI would recommend that you by it.	It s really hard for me to explain what it is that is so great about the series.
4	I m not sure how something can be so good, it can t belong in a group.	I can t imagine a fantasy fan who hasn t read this yet, but I have heard of them.

1. [14411]
   - ⊕ Even if the characters were just talking, I was really interested in what was said.
   - ⊕ I'm not sure how something can be so good, it can't belong in a group.
   - ⊕ The Lord of The Rings is a series that you will want to read over and over, just to see all of the characters again.
   - ⊖ -TitleOne of the best series ever! -Rating5.0 -ProsIt's THE LORD OF THE RINGS! -ConsThere are only 4 books in the series -The Bottom LineIf you like fantasy, or are a complete newbie, you will need to read Lord of the Rings. -Full Review-

Figure 8.7: Ooops! contrastive opinion summary generation screen shot

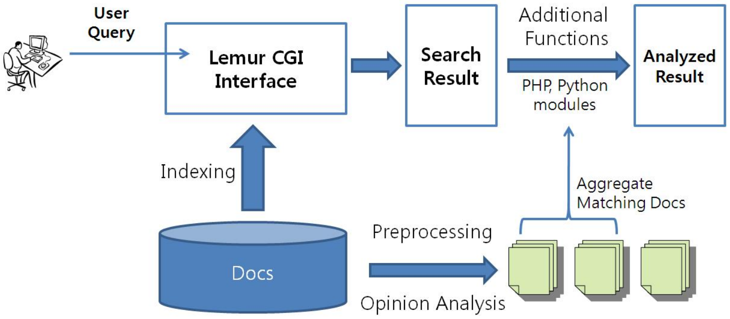


Figure 8.8: Ooops! system architecture

processed document. After the opinion polarity preprocessing, each document has opinion sentences with the polarity marking. When we trigger the opinion analysis function, the javascript extracts the list of retrieved document numbers on the current search result page and passes it to the php module. The php module retrieves already processed opinion sentences by matching document numbers. Because we just merge all the result without on-the-fly opinion analysis, our system does not have delay for opinion sentence retrieval. For the contrastive opinion summary generation, we use Python module as we do for contrastive opinion summary generation system in Section 8.1. Because contrastive opinion summarization needs different computation depending on the input sentences, the final generation step should be executed when a user request arrives.

Chapter 9

Related Work

Opinion summarization is an active research area because of the increased volume of opinionated data. General opinion mining was focused on finding topics among articles and clustering positive and negative opinions on topics [7, 8, 13, 9, 20, 16]. Most of the results of opinion summarization focused on showing statistics of the number of positive and negative opinions. Usually people used table-shaped summary [7, 8, 16] or histogram [13]. Sometimes, each section had an extracted sentence from the article and had a link to the original one. It was not enough to show the details of the different opinions. Representative opinion summarization works mainly focused on sentiment classification on various aspects [7, 8, 13]. These studies generally relied on heuristic methods and data mining techniques such as association rule mining to identify aspects and sentiments of aspects.

Some opinion summarization work used probabilistic topic modeling methods such as probabilistic latent semantic analysis (PLSA) [6] and latent Dirichlet allocation (LDA) [3]. Topic sentiment mixture model [16] extended PLSA model with opinion priors to show positive and negative aspects of topics effectively. This model finds latent topics as well as its associated sentiment and also reveals how opinion sentiments evolve over the time line. In [22], multi-grain topic model was proposed as an extension of LDA. This work finds ratable aspects from reviews and generates summaries for each aspect. The proposed multi-grain LDA topic model can extract local topics which are ratable aspects written by an individual user as well as cluster local topics into global topics of objects such as the brand of a product type.

Heuristic rule-based methods have also been used in opinion summarization. Usually these methods have two steps: features extraction and opinion finding for each feature. In [13, 8, 9], features of products are found using supervised association rule mining and rules such as opinion features are usually noun phrases. To connect extracted features with opinion words, WordNet is also used. [24] focused on movie review domain. Based on domain-specific heuristics such as many features tend to be around the cast of a movie, features can be found more efficiently. Machine learning techniques [18, 11] and relaxation labeling [20] are also used for features extraction and opinion summary.

In addition to these representative probabilistic and rule-based approaches to opinion summarization, opinion integration [14] and sentiment classification [17] are also related to our work.

A main distinction of our work from these studies is that we aim at summarizing contradictory opinions. As discussed in the beginning of this thesis, our work extends the existing work on opinion summarization to help users further digest and understand contradictory opinions.

There were some works on extracting comparative sentences or detecting contradiction in text. In [10], methods are proposed for detecting comparative sentences by checking signal words like ‘than’. In [5], the authors structurally analyzed the characteristics of contradiction and suggested heuristic methods for detecting contradiction in text. Some work considered finding contradiction as a binary classification problem. In [21], support vector machine(SVM) is applied on classification of support and opposition in text, while in [2, 15], methods based on graph representing relationship between texts or authors are used for classifying texts. Although these works proposed methods to find contradictions, they did not directly address the problem of summarizing contradictory opinions, for which we need to model both contrastiveness and representativeness.

In [4], the authors studied visualization of different opinions and showed various visualization methods using graph and tree. The work [23] about mining mixed opinions using topic model is also related to the current work. But none of these works can generate a comparative summary of contradictory opinions as we do.

Different sentence similarity measures are explored in [1]. The authors compared the performance of word overlap measures, TF-IDF measures, linguistic measures and combination of them over different data corpus, and found that linguistic measures perform the best in finding similar sentences, and TF-IDF measures perform well for deciding whether input sentences are dissimilar or not. We also compared different similarity measures for the COS problem, and our results show that semantic matching appears to be not useful for this task.

Chapter 10

Conclusion

In this thesis, we proposed a novel summarization problem, namely contrastive opinion summarization. It aims to summarize contradictory or mixed opinions about a topic and generate a list of contrastive pairs of sentences with different sentiment polarities to help users to digest contradictory opinions. We formally framed the problem as an optimization problem and proposed two approximation methods to solve the optimization problem. We also explored different similarity measures in our optimization framework. We leverage existing summarization resources to create a gold standard data set for evaluating the proposed new summarization task.

Experiment results using this data set show that the proposed methods are effective for generating contrastive opinion summaries. In particular, contrastiveness-first approximation works better than representativeness-first, and the heuristic of removing sentimental words in computing contrastive similarity is effective. However, semantic term matching based on WordNet is found to be not helpful. Sample summary results show that the generated summaries are informative and can help users digest contradictory opinions more effectively.

In addition to experiments, we implemented two demo systems which allow users to apply the proposed method to their own data and show a possible search engine using the proposed method.

Our work can be extended in several directions. First, more experiments on additional larger data sets would be desirable. Second, we have only explored some basic semantic term matching method; it would be interesting to further explore more advanced similarity functions such as those based on sentence alignment or more in-depth semantic analysis. Third, it should also be very interesting to further study how to develop algorithms to achieve better approximate solutions to the optimization problem using our framework. Finally, improving demo system and using user feedback is also very useful for future exploring opinion summarization research. Better demo system attracts more users and their feedback. This will help to further improve the system and can lead to the new research topics.

References

[1] P. Achananuparp, X. Hu, and X. Shen. The evaluation of sentence similarity measures. In *DaWaK '08: Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery*, pages 305–316, Berlin, Heidelberg, 2008. Springer-Verlag.

[2] M. Bansal, C. Cardie, and L. Lee. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. In *Proceedings of COLING: Companion volume: Posters*, pages 13–16, 2008.

[3] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

[4] C. Chen, F. I. Sanjuan, E. Sanjuan, and C. Weaver. Visual analysis of conflicting opinions. In *IEEE Symposium on Visual Analytics Science and Technology (VAST 2006).*

[5] M. C. de Marneffe, A. N. Rafferty, and C. D. Manning. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[6] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, New York, NY, USA, 1999. ACM.

[7] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM.

[8] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of AAAI*, pages 755–760, 2004.

[9] M. Hu and B. Liu. Opinion extraction and summarization on the Web. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-2006), Nectar Paper Track*, Boston, MA, 2006.

[10] N. Jindal and B. Liu. Identifying comparative sentences in text documents. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 244–251, New York, NY, USA, 2006. ACM.

[11] L.-W. Ku, Y.-T. Liang, and H.-H. Chen. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 100–107, 2006.

[12] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications).* Springer, January 2007.

[13] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 342–351, New York, NY, USA, 2005. ACM Press.

[14] Y. Lu and C. Zhai. Opinion integration through semi-supervised topic modeling. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 121–130, New York, NY, USA, 2008. ACM.

[15] R. Malouf and T. Mullen. Graph-based user classification for informal online political discourse. In *Proceedings of the 1st Workshop on Information Credibility on the Web (WICOW)*, Miyazaki, Japan, 2007.

[16] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 171–180, New York, NY, USA, 2007. ACM.

[17] B. Pang and L. Lee. *Opinion Mining and Sentiment Analysis*, volume 2(1–2) of *Foundations and Trends in Information Retrieval*. Now Publ., 2008.

[18] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[19] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet: : Similarity - measuring the relatedness of concepts. In *AAAI*, pages 1024–1025, 2004.

[20] A.-M. Popescu and O. Etzioni. Extracting Product Features and Opinions from Reviews. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 339–346. Association for Computational Linguistics, October 2005.

[21] M. Thomas, B. Pang, and L. Lee. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 327–335, 2006.

[22] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 111–120, New York, NY, USA, 2008. ACM.

[23] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 743–748, New York, NY, USA, 2004. ACM.

[24] L. Zhuang, F. Jing, X. Zhu, and L. Zhang. Movie review mining and summarization. In *Proceedings of the ACM SIGIR Conference on Information and Knowledge Management (CIKM)*, 2006.

[25] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.