# Answering Opinion Questions on Products by Exploiting Hierarchical Organization of Consumer Reviews

**Jianxing Yu, Zheng-Jun Zha, Tat-Seng Chua**
School of Computing
National University of Singapore
{jianxing, zhazj, chuats}@comp.nus.edu.sg

## Abstract

This paper proposes to generate appropriate answers for opinion questions about products by exploiting the hierarchical organization of consumer reviews. The hierarchy organizes product aspects as nodes following their parent-child relations. For each aspect, the reviews and corresponding opinions on this aspect are stored. We develop a new framework for opinion Questions Answering, which enables accurate question analysis and effective answer generation by making use the hierarchy. In particular, we first identify the (explicit/implicit) product aspects asked in the questions and their sub-aspects by referring to the hierarchy. We then retrieve the corresponding review fragments relevant to the aspects from the hierarchy. In order to generate appropriate answers from the review fragments, we develop a multi-criteria optimization approach for answer generation by simultaneously taking into account review salience, coherence, diversity, and parent-child relations among the aspects. We conduct evaluations on 11 popular products in four domains. The evaluated corpus contains 70,359 consumer reviews and 220 questions on these products. Experimental results demonstrate the effectiveness of our approach.

## 1 Introduction

With the rapid development of E-commerce, most retail websites encourage consumers to post reviews to express their opinions on the products. For example, the review "*The battery of Nokia N95 is amazing.*" reveals positive opinion on the aspect "*battery*" of product *Nokia N95*. An *aspect* here refers to a component or an attribute of a certain product. Numerous consumer reviews are now available online, and these reviews contain rich opinionated information on various aspects of products. They are naturally a valuable resource for answering opinion questions about products, such as "*How do people think about the battery of Nokia N95?*" Opinion Question Answering (opinion-QA) on products seeks to uncover consumers' thinking and feeling about the products or aspects of products. It is different from traditional factual QA, where the questions ask for the fact, such as "*Where is the capital of United States?*" and the answer is "*Washington, D.C.*"

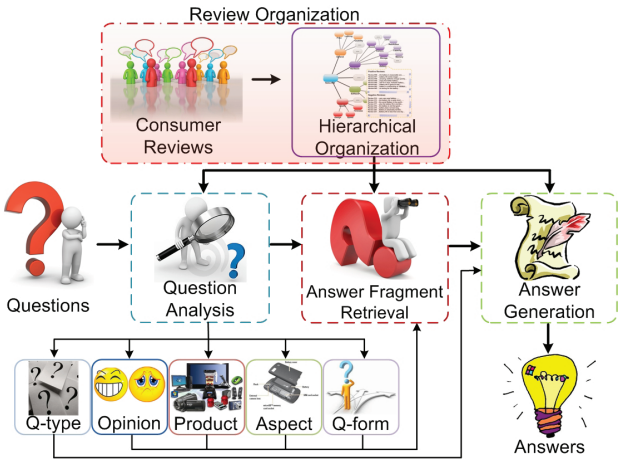


Figure 1: Overview of product opinion-QA framework

For a product opinionated question, the answer should not be just a best answer. It should reflect the opinions of various segments of users, and incorpo-



*Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 391–401, Jeju Island, Korea, 12–14 July 2012. 

rate both positive and negative viewpoints. Hence the answer should be a summarization of public opinions and comments on the product or specific aspect asked in the question (Jiang et al., 2010). In addition, it should also include public opinions and comments on the sub-aspects. Such answers would help users to understand the inherent reasons of the opinions on the asked aspect. For example, the question "*What do people think the camera of Nokia 5800?*" asks for public positive and negative opinions on the aspect "*camera*" of product "*Nokia 5800*." The summarization of opinions on the sub-aspects such as "*lens*" and "*resolution*" would help users better understand that the public complaints on the aspect "*camera*" are due to the poor "*lens*" and/or low "*resolution*." Moreover, the answer should be presented following the general-to-specific logic, i.e., from general aspects to specific sub-aspects. This makes the answer easier to understand by the users (Ouyang et al., 2009).

Current Opinion-QA methods mainly include three components, including question analysis that identifies aspects and opinions asked in the questions, answer fragment retrieval, and answer generation which summarizes the retrieved fragments (Lloret et al., 2011). Although existing methods show encouraging performance, they are usually not able to generate satisfactory answers due to the following drawbacks. First, current methods often identify aspects as the noun phrases in the questions. However, noun phrases contain noises that are not aspects. This gives rise to imprecise aspect identification. For example, in the question "*What reasons can I persuade my wife that people prefer the battery of Nokia N95?*" noun phrases "*wife*" and "*people*" are not aspects. Moreover, current methods relied on noun phrases are not able to reveal the implicit aspects, which are not explicitly asked in the questions. For example, the question "*Is iPhone 4 expensive?*" asks about the aspect "*price*", but the term "*price*" does not appear in the question. Second, current methods cannot discover sub-aspects of the asked aspect due to its ignorance of parent-child relations among aspects. Third, the answers generated by the existing methods do not follow the general-to-specific logic, leading to difficulty in understanding the answers.

To overcome these problems, we can resort to the hierarchical organization of consumer reviews on products. As illustrated in Figure 2, the hierarchy organizes product aspects as nodes, following their parent-child relations. For each aspect, the reviews and corresponding opinions on this aspect are stored. Such hierarchy can naturally facilitate to identify aspects asked in questions. While explicit aspects can be recognized by referring to the hierarchy, implicit aspects can be inferred based on the associations between sentiment terms and aspects in the hierarchy (Yu et al., 2011). The sentiment terms are discovered from the reviews on corresponding aspects. Moreover, by following the parent-child relations in the hierarchy, sub-aspects of the asked aspect can be directly acquired, and the answers can present aspects from general to specific.

Motivated by the above observations, we propose to exploit the hierarchical organization of consumer reviews for product opinion-QA. As illustrated in Figure 1, our framework first organizes consumer reviews of a certain product into a hierarchical organization. The resulting hierarchy is in turn used to help question analysis and relevant review fragments retrieval. In order to generate appropriate answers from the retrieved fragments, we develop a multi-criteria optimization approach by simultaneously taking into account review salience, coherence, and diversity. The parent-child relations among aspects are also incorporated into the approach to ensure the answers be general-to-specific. We conduct evaluations on 11 popular products in four domains. The evaluated corpus contains 70,359 consumer reviews and 220 questions on these products. More details of the dataset are discussed in Section 4. Experimental results to demonstrate the effectiveness of our approach.

The main contributions of this paper include,

- We propose to exploit the hierarchical organization of consumer reviews for answering opinion questions on products.
- With the help of the hierarchy, our proposed framework can accurately identify (explicit/implicit) aspects asked in the questions, and the corresponding sub-aspects.
- We develop a multi-criteria optimization approach to generate informative, coherent, diverse and general-to-specific answers.

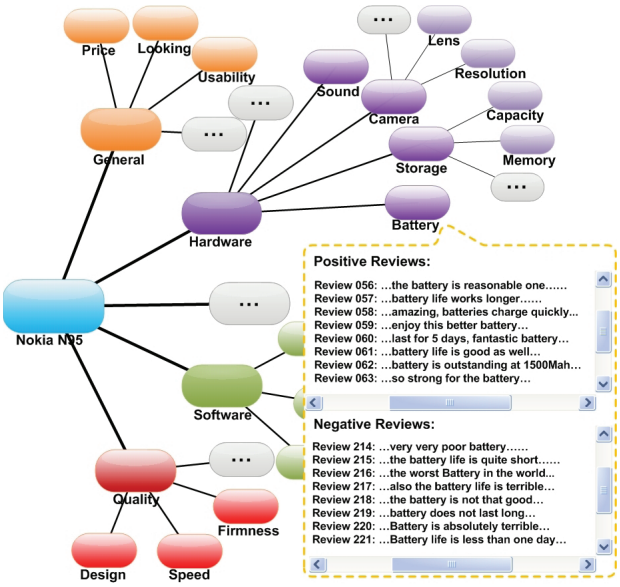


Figure 2: Hierarchical organization for *Nokia N95*

The rest of this paper is organized as follows. Section 2 introduces the components of hierarchical organization of reviews, question analysis, and answer fragment retrieval. Section 3 elaborates the multi-criteria optimization approach for answer generation . Section 4 presents experimental details, while Section 5 reviews related works. Finally, Section 6 concludes this paper with future works.

## 2 Hierarchical Organization, Question Analysis, and Answer Fragment Retrieval

Let $\mathcal{R} = \{r_1, \cdots, r_{|\mathcal{R}|}\}$ denote a collection of consumer reviews of a certain product. Each review reflects consumer opinions on the product and/or product aspects. Let $q$ denote an opinion question, which asks for public opinions on a product or some aspects of the product. The task is to retrieve the opinionated review fragments relevant to the asked product/product aspects, and summarize these fragments to form an appropriate answer to question $q$.

Next, we introduce the components of hierarchical organization that organizes consumer reviews into a hierarchy, question analysis which identifies the products/aspects and opinions asked in the questions, and answer fragment retrieval that retrieves review fragments relevant to the questions.

### 2.1 Hierarchical Organization of Reviews

We employ the method proposed by Yu et al. (2011) to organize consumer reviews of a product into a hierarchical organization. As shown in Figure 2, the hierarchy organizes product aspects as nodes, following their parent-child relations. In particular, this method first automatically acquires an initial aspect hierarchy from the domain knowledge and identifies aspects commented in the reviews. It then incrementally inserts the identified aspects into appropriate positions in the initial hierarchy, and finally obtains an aspect hierarchy that allocates all the newly identified aspects. The consumer reviews are then organized to their corresponding aspect nodes in the hierarchy. Sentiment classification is then performed to determine consumer opinions on the reviews.

The reported performance of Yu et al. (2011) on aspect identification, aspect hierarchy generation and sentiment classification are 0.731, 0.705, 0.787 in terms of average $F_1$-measure, respectively.

### 2.2 Question Analysis and Answer Fragment Retrieval

Question analysis consists of five sub-tasks: recognizing product asked in the question; identifying aspects in the question; classifying opinions that the question asks for (the asked opinion could be positive, negative or both); identifying the question type (e.g. asking for public opinions, or the reason of the opinions, etc.); and identifying the question form (i.e. comparative question or single form question).

***Recognizing the product***: A name entity recognizer [1] is trained to recognize the product name. In particular, we collect 420 auxiliary questions from Yahoo!Answer [2], and manually annotate the product names (submitted as supplementary material in Appendix A). A name entity recognizer for product is learned on these data, with unigrams and POS tags as features. Given a testing question, the recognizer predicts each word as *B*, *I*, *E* or *O*, where *B, I, E* denote the begin, internal, and end of a product name respectively, and *O* corresponds to other words.

***Identifying aspects***: As aforementioned, simply extracting the noun phrases as aspects would import noises. Also, some "implicit" aspects do not ex-

[1] http://nlp.stanford.edu/software/CRF-NER.shtml
[2] http://answers.yahoo.com

plicitly appear in the reviews. One simple solution for these problems can resort to the review hierarchy. The hierarchy has organized product aspects, which can be used to filter the noise noun phrases for accurately identifying the explicit aspects. For the implicit aspects, we observe they are usually modified by some peculiar sentiment terms (Su et al., 2008). For example, the aspect "*size*" is often modified by the sentiment terms such as "*large*", but seldom by the terms such as "*expensive*." Thus, there are some associations between the aspects and sentiment terms. Such associations can be learned from the hierarchy and leveraged to infer the implicit aspects (Yu et al., 2011). In order to simultaneously identify the (explicit/implicit) aspects, we adopt a hierarchical classification technique. The technique simultaneously learns to identify explicit aspects, and discovers the associations between aspects and sentiment terms by multiple classifiers. In particular, given a testing question, we identify its aspect by hierarchically classify (Silla et al., 2011) it into the appropriate aspect node of a particular product hierarchy. The classification greedily searches a path in the hierarchy from top to down. The search begins at the root node, and stops at the leaf node or a specific node where the relevance score is lower than a pre-defined threshold. The relevance score on each node is determined by a SVM classifier. Multiple SVM classifiers are learned on the hierarchy, one distinct classifier for a node. The reviews that are stored in the node and its child-nodes are used as training samples. We employ the features of noun terms, and sentiment terms in the sentiment lexicon provided by *MPQA* project (Wilson et al., 2005).

***Classifying the opinions***: Given a set of testing questions, we first distinguish the opinion questions from the factual ones (Yu et al., 2003). Since the opinion questions often contain one or more sentiment terms, we classify them by employing the sentiment terms in the sentiment lexicon provided from MPQA project (Wilson et al., 2005). Subsequently, we learn a SVM sentiment classifier to determine the opinion polarity of the opinion questions. In particular, the reviews and corresponding opinions stored in the hierarchy are used as training samples, which are represented by the unigram features.

***Identifying the question type***: Opinion questions are often categorized into four types (Ku et al., 2007),

- **Attitude** question, asking for public opinion on a product or product aspect, such as "*What do people think iPhone 3gs?*"
- **Reason** question, asking for the reason of public opinion on a product or product aspect, such as "*Why do people like iPhone 3gs?*"
- **Target** question, asking for the object in the public opinion, such as "*Which phone is better than Nokia N95?*"
- **Yes/No** question, asking for whether a statement is correct, such as "*Is Nokia N95 bad?*"

We formulate the question type identification as a multi-class classification problem. A multi-class SVM classifier [3] is learned for the classification. We collect 420 auxiliary questions from Yahoo!Answer and manually annotate their types (submitted as supplementary material in Appendix B). These questions are used for training, with POS tags and question words (i.e. why, what, how, do, is) as features.

***Identifying the question form***: Question form includes single and comparative. A question is viewed as comparative if it contains comparative adjectives and adverbs (e.g. cheaper, etc.), otherwise as the single form (Moghaddam et al., 2011). The POS tags are exploited to detect comparative adjectives (i.e. tag "*JJR*") and adverbs (i.e. tag "*RBR*").

After analyzing the question, we retrieve all review sentences on the asked aspect and all its subaspects from a certain product hierarchy, and choose the ones relevant to the opinion asked in the question. For the single form question, we view the retrieved sentences as the answer fragments. For the comparative questions, we select comparative sentences on the compared products from the retrieved sentences, and treat them as the answer fragments. Subsequently, question type is used to define the template for the answers. In particular, for the questions asking for reason and attitude, we generate the answers by summarizing corresponding answer fragments. For questions seeking for a target as the answer, we output the product names based on the majority voting of the opinions in the retrieved answer fragments. For the yes/no questions, we first generate the "yes/no" answer based on the

[3] http://svmlight.joachims.org/svm_multiclass.html

consistency between the asked opinions and the major opinions in the answer fragments, and then summarize these fragments to form the answers.

## 3 Answer Generation

Answer generation aims to generate an appropriate answer for a given opinion question based on the retrieved answer fragments, i.e., review sentences. An answer is essentially a sequence of sentences. Hence, the task of answer generation is to select sentences from the retrieved answer fragments and order them appropriately. We formulate this task into a multi-criteria optimization problem. We incorporate multiple criteria in the answer generation process, including answer salience, coherence, and diversity. The parent-child relations between aspects is also incorporated to ensure the answer follow the general-to-specific logic. In the next subsections, we will introduce details of the proposed multi-criteria optimization approach.

### 3.1 Formulation

We first introduce the multiple criteria and then present the optimization problem.

***Salience*** is used to measure the representativeness of the answer. A good answer should consist of salient review sentences. Let $\mathcal{S}$ denote the set of retrieved sentences. We define a binary variable $s_i \in \{0, 1\}$ to indicate the selection of sentence $i$ for the answer, i.e. $s_i$ = 1 (or 0) indicates that $s_i$ is selected (or not). Let $\omega_i$ denote the salience of sentence $i$. The estimation of $\omega_i$ will be described in Section 3.2. The salience score of the answer (i.e., a set of sentences) is computed by summing up the scores of all its constituent sentences, as $\sum_{i\in\mathcal{S}} \omega_i s_i$.

***Coherence*** is used to quantify the readability of an answer. To make the answer readable, the constituent sentences in the answer should be ordered properly. That is, the adjacent sentences should be coherent. We define $e_{i,j} \in \{0, 1\}$ to indicate whether the sentences $i$ and $j$ are adjacent in the answer; where $e_{i,j}$ = 1 (or 0) means they are (or not) adjacent. The coherence between two adjacent sentences is measured by $c_{ij}$. The estimation of $c_{ij}$ will be described in Section 3.3. As aforementioned, the answer is expected to be presented in a general-to-specific manner, i.e. from general aspects to specific sub-aspects. We define $h_{i,j}$ in Eq.1 to measure the general-to-specific coherence of sentences $i$ and $j$.

$$h_{i,j} = \begin{cases} e^{-\frac{1}{level_i - level_j}}; & \text{if } level_i \neq level_j; \\ 1; & \text{otherwise,} \end{cases} \quad (1)$$

where $level_i$ denotes level position of the aspect commented in sentence $i$ by referring to the hierarchy, with the root level being 0. The coherence score of the answer is computed by summing up the scores of all its adjacent sentences as, $\sum_{j\in\mathcal{S}} \sum_{i\in\mathcal{S}} h_{i,j} c_{i,j} e_{i,j}$.

***Diversity.*** A good answer should diversely cover all the important information. We introduce a matrix $\mathcal{M}$ in Eq.2 to measure the pairwise diversities among sentences. $\mathcal{M}_{ij}$ corresponds to the diversity between sentences $i$ and $j$. When sentences $i$ and $j$ comment on the same aspects, $\mathcal{M}_{ij}$ will favor to select the pair of sentences that discusses on diverse content (i.e. low similarity). Otherwise, the pair of sentences commented on different aspects is viewed to be diverse, and $\mathcal{M}_{ij}$ is set as a constant bigger than one.

$$\mathcal{M}_{ij} = \begin{cases} 1 - similar(i, j) & if\ i, j\ commented\ on\ same\ aspect \\ \varphi & otherwise, \end{cases} \quad (2)$$

where $\varphi$ is a constant [4].

***Multi-Criteria Optimization*** We integrate the above criteria into the multi-criteria optimization formulation,

$$\begin{cases} \max\{\lambda_1 \cdot \sum_{i\in\mathcal{S}} \omega_i s_i + \lambda_2 \cdot \sum_{j\in\mathcal{S}} \sum_{i\in\mathcal{S}} h_{i,j} c_{i,j} e_{i,j} \\ \quad + \lambda_3 \cdot \sum_{j\in\mathcal{S}} \sum_{i\in\mathcal{S}} s_i \mathcal{M}_{ij}; \\ s_i, e_{i,j} \in \{0, 1\}, \forall i, j; \\ \lambda_1 + \lambda_2 + \lambda_3 = 1,\ 0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1, \end{cases} \quad (3)$$

where $\lambda_1$, $\lambda_2$, $\lambda_3$ are the trade-off parameters.

We further incorporate the following constrains into the optimization framework, so as to derive appropriate answers.

- The length of the answer is up to $K$,

$$\sum_{i\in\mathcal{S}} l_i s_i \leq K, \quad (4)$$

  where $l_i$ is the length of sentence $i$.
- When sentence $i$ is not selected (i.e. $s_i = 0$), the adjacency between any sentence to $i$ is set

[4] Empirically set to 10 in the experiment.

to zero (i.e. $\sum_{i\in\mathcal{S}} e_{i,j} = \sum_{i\in\mathcal{S}} e_{j,i} = 0$). When sentence $i$ is selected, there are two sentences adjacent to sentence $i$ in the answer, one before $i$ and another after $i$. (i.e. $\sum_{i\in\mathcal{S}} e_{i,j} = \sum_{i\in\mathcal{S}} e_{j,i} = 1$).

$$\sum\nolimits_{i\in\mathcal{S}} e_{i,j} = \sum\nolimits_{i\in\mathcal{S}} e_{j,i} = s_j, \quad \forall j. \tag{5}$$

- In order to avoid falling into a cycle in sentence selection, we employ the following constraints (Deshpande et al., 2009).

$$\begin{array}{l} \sum_{i\in\mathcal{S}} f_{0,i} = n+1; \\ \sum_{i\in\mathcal{S}} f_{i,n+1} \geq 1; \\ \sum_{i\in\mathcal{S}} f_{i,j} - \sum_{i\in\mathcal{S}} f_{j,i} = s_j, \quad \forall j; \\ 0 \leq f_{i,j} \leq (n+1)\cdot e_{i,j}, \ \forall i,j, \end{array} \tag{6}$$

where the variable $f_{i,j}$ is an integer to number the selected adjacent sentences from 1 to n+1, and the first selected sentence is numbered $f_{0,i}$ = n + 1. If the last selected sentence obtains a number $f_{i,n+1}$ which is bigger then 1, then the selection has no cycle.

***Solution***

Given the salience weights $\omega_i|_{i=1}^{\mathcal{S}}$, and coherence weights $c_{i,j}|_{i,j=1}^{\mathcal{S}}$, the above multi-criteria optimization problem can be solved by *Integer Linear Programming* (Schrijver et al., 1998). The optimal solutions $s_i|_{i=1}^{\mathcal{S}}$ and $e_{i,j}|_{i,j=1}^{\mathcal{S}}$ indicate the selected sentences and the order of them. In the next subsections, we will introduce the estimations of $\omega_i|_{i=1}^{\mathcal{S}}$ and $c_{i,j}|_{i,j=1}^{\mathcal{S}}$.

### 3.2 Salience Weight Estimation

The salience weight of sentence $i$ is formulated as $\omega_i = \sum_{g=1}^{G} \varphi_g(i)/G$, where $\varphi(i)$ denotes the measurement for the importance of sentence $i$. We define seven measurements (i.e. $G = 7$) below.

***Helpfulness***: Many forum websites provide a helpfulness score, which is used to rate the quality of a review. The sentences that come from helpful reviews are often representative (Mizil et al., 2009). We compute $\varphi(i)$ of sentence $i$ by using helpfulness score from its host review.

***Timeliness***: The new coming sentence often contains more updated and useful information (Liu et al., 2008). $\varphi(i)$ is the post time of sentence $i$. We normalize it to $[0, 1]$.

***Grammaticality***: The grammatical sentence is often more readable. We employ the method in Agichtein et al. (2008) to calculate the grammar score. In particular, $\varphi(i)$ is calculated by the KL-divergence between language models of sentence $i$ to Wikipedia articles.

***Position***: The first sentence in a review is usually informative (He et al., 2011). $\varphi(i)$ is computed based on the position of the sentence in the review, i.e. $\varphi(i) = 1/position_i$.

***Aspect Frequency***: The sentence that contains the frequent aspects is often salient (Nishikawa et al., 2010). Hence, $\varphi(i)$ is computed as the sum of the frequency for aspects in sentence $i$.

***Centroid Distance***: As aforementioned, review sentences are stored in the corresponding aspect nodes in the hierarchy. The sentence that is close to the centroid of the reviews stored in an aspect node is more likely to be salient (Erkan et al., 2004). $\varphi(i)$ is computed as the Cosine similarity between sentence $i$ to the corresponding review cluster centroid based on the unigram features.

***Local Density***: The sentence would be informative when it is in the dense part of the aspect node in the feature space (Scott et al., 1992). We employ *Multivariate Kernel Density Estimation* to estimate the density. We first represent all the sentences stored in each node into feature vectors, with unigram as features. The density of a sentence is then calculated as $\varphi(\boldsymbol{x}) = \sum_{i=1}^{n} K_H(\boldsymbol{x} - \boldsymbol{x_i})/n$, where $\boldsymbol{x}$ denotes the feature vector of sentence $i$, $n$ is the size of sentences stored in the node, and $K_H(\boldsymbol{x}) = (2\pi)^{-1/2}\exp(-1/2(\boldsymbol{x}^T\boldsymbol{x}))$ represents the *Gaussian* kernel.

### 3.3 Coherence Weight Estimation

The coherence $c_{i,j}$ between sentences $i$ and $j$ is formulated as $c_{i,j} = \boldsymbol{\mu}\cdot\boldsymbol{\psi(i,j)}$, where $\boldsymbol{\mu}$ is a weight vector, and $\boldsymbol{\psi(i,j)}$ denotes the feature function. $\boldsymbol{\psi(i,j)}$ takes two sentences $i$ and $j$ as input, and outputs a vector with each dimension indicating the present/absent of a feature. In order to capture the sequential relations among sentences, we utilize features as the *Cartesian* product over the terms of N-gram (N=1,2) and POS tags generated from sentences $i$ and $j$ (Lapata et al., 2003).

To learn the weight vector $\boldsymbol{\mu}$, we employ the *Passive-Aggressive* algorithm (Crammer et al.,

2006). It is an online learning algorithm, so that we can update the weight when more consumer reviews are available. The algorithm takes up one training sample and outputs the solution that has the highest score under the current weight. If the output differs from training samples, the weight vector is updated according to Eq.7. Since the consumer reviews often include multiple sentences, we can directly use the adjacency of these sentences as training samples. In particular, we treat the adjacent sentence pairs in the reviews as training samples (i.e. $c_{i,j} = 1$).

$$\begin{aligned} &\min \|\boldsymbol{\mu}^{\boldsymbol{i+1}} - \boldsymbol{\mu}^{\boldsymbol{i}}\| \\ &\begin{cases} \boldsymbol{\mu}^{\boldsymbol{i+1}} \cdot \boldsymbol{\Psi}(\mathbf{p}, \mathbf{q}^*) - \boldsymbol{\mu}^{\boldsymbol{i+1}} \cdot \boldsymbol{\Psi}(\mathbf{p}, \hat{\mathbf{q}}) \geq \tau(\hat{\mathbf{q}}, \mathbf{q}^*); \\ \tau(\hat{\mathbf{q}}, \mathbf{q}^*) = \frac{2 \cdot T(\hat{\mathbf{q}}, \mathbf{q}^*)}{m(m-1)/2}, \end{cases} \end{aligned} \tag{7}$$

where $\boldsymbol{\mu}^{\boldsymbol{i}}$ is the current weight vector and $\boldsymbol{\mu}^{\boldsymbol{i+1}}$ is the updated vector, $\mathbf{q}^*$ and $\hat{\mathbf{q}}$ are the gold standard and predicted sequence of sentences, respectively, $\mathbf{p}$ denotes a set of sentences, $\boldsymbol{\Psi}(\cdot)$ is the feature function on the whole feature space (i.e. $\sum \boldsymbol{\psi}(\cdot)$), $\tau(\cdot, \cdot)$ is a *Kendall's tau* lost function (Lapata et al., 2006), $T(\cdot, \cdot)$ represents the number of inversion operations that needs to bring $\hat{\mathbf{q}}$ to $\mathbf{q}^*$, and $m$ denotes the number of sentences.

## 4 Evaluations

In this section, we evaluate the effectiveness of the proposed approach, in terms of question analysis and answer generation.

### 4.1 Data Set and Experimental Settings

We employed the product review dataset used in Yu et al. (2011) as corpus. As illustrated in Table 1, the dataset contained 70,359 reviews about 11 popular products in four domains. In addition, we created 220 questions for these products by referring to real questions in Yahoo!Anwser service. We corrected the typos and grammar errors for these real questions. Each product contains 15 opinion questions and 5 factual questions, respectively. All questions were shown in Appendix C in supplementary material. Three annotators were invited to generate the gold standard. Each question was labeled by two annotators. The labels include product name, product aspect, opinion, question type and question form. The average inter-rater agreement in terms of Kappa statistics is 89%. These annotators were then invited to read the reviews, and create the ground truth answers by selecting and ordering some review sentences. Such process is time consuming and labor-intensive. We speed up the annotation process as follows. We first collected all the review sentences in the answers generated by three evaluated methods to be discussed in Section 4.3.1. In addition, we sampled the top-N (N=20) sentences on each asked aspect and its sub-aspects respectively, where the sentences were ranked based on their salient weights in Section 3.2. We then provided such subset of review sentences to the three annotators, and let them individually create an answer of up to 100 words (i.e. K=100) for each question.

| *Product Name* | *Domain* | *Review#* | *Sentence#* |
|---|---|---|---|
| Canon EOS 450D (Canon EOS) | camera | 440 | 628 |
| Fujifilm Finepix AX245W (Fujifilm) | camera | 541 | 839 |
| Panasonic Lumix DMC-TZ7 (Panasonic) | camera | 650 | 1,546 |
| Apple MacBook Pro (MacBook) | laptop | 552 | 4,221 |
| Samsung NC10 (Samsung) | laptop | 2,712 | 4,946 |
| Apple iPod Touch 2nd (iPod Touch) | MP3 | 4,567 | 10,846 |
| Sony NWZ-S639 16GB (Sony NWZ) | MP3 | 341 | 773 |
| BlackBerry Bold 9700 (BlackBerry) | phone | 4,070 | 11,008 |
| iPhone 3GS 16GB (iPhone 3GS) | phone | 12,418 | 43,527 |
| Nokia 5800 XpressMusic (Nokia 5800) | phone | 28,129 | 75,001 |
| Nokia N95 | phone | 15,939 | 44,379 |

Table 1: Statistics of the product review dataset, # denotes the number of the reviews/sentences.

We employed *precision* (P), *recall* (R) and $F_1$-measure ($F_1$) as the evaluation metric for question analysis, and utilized *ROUGE* (Lin et al., 2003) as the metric to evaluate the quality of answer generation. *ROUGE* is a widely accepted standard for summarization, which measures the quality of the summarized answers by counting the overlapping N-grams between the answers generated by machine and human, respectively. In the experiment, we reported the $F_1$-measure of *ROUGE-1*, *ROUGE-2* and *ROUGE-SU4*, which count the overlapping unigrams, bigrams and skip-4 bigrams respectively. *ROUGE-1* can measure informativeness of the answers, while higher order *ROUGE-N* (*N=2,4*) captures the matching of subsequences, which can measure the fluency and readability of the answers. For the trade-off parameters, we empirically set $\lambda_1 = 0.4$, $\lambda_2 = 0.3$ and $\lambda_3 = 0.3$.

### 4.2 Evaluations on Question Analysis

We first evaluated the performance of product recognition, opinion/factual question classification, opinion classification, question type and question form identification. The experimental results are shown

in Table 2. The results show that traditional methods achieve encouraging performance on the aforementioned tasks.

| *Evaluated Topics* | $P$ | $R$ | $F_1$ |
|---|---|---|---|
| Product recognition | 0.755 | 0.618 | 0.680 |
| Opinion/factual | 0.897 | 0.895 | 0.893 |
| Opinion classification | 0.755 | 0.745 | 0.748 |
| Question type | 0.800 | 0.775 | 0.783 |
| Question form | 0.910 | 0.903 | 0.905 |

Table 2: Performance of question analysis.

| *Methods* | $P$ | $R$ | $F_1$ |
|---|---|---|---|
| Our method | **0.851*** | **0.763*** | **0.805*** |
| Balahur's method | 0.825 | 0.400 | 0.538 |

Table 3: Performance of aspect identification for question analysis. * denotes the results (i.e. $P$, $R$, $F_1$) are tested for statistical significance using T-Test, p-values$<$0.05.

| *Methods* | $P$ | $R$ | $F_1$ |
|---|---|---|---|
| Our method | **0.726*** | **0.643*** | **0.682*** |
| Su's method | 0.689 | 0.571 | 0.625 |

Table 4: Performance of implicit aspect identification for question analysis. T-Test, p-values$<$0.05

We next examined the performance of our approach on aspect identification. The method proposed by Balahur et al. (2008) was reimplemented as the baseline, which identifies aspects based on noun phrase extraction. This method achieved good performance on the opinion QA task in TAC 2008 and was employed in subsequent works. As demonstrated in Table 3, our approach significantly outperforms Balahur's method by over 49.4% in terms of average $F_1$-measure. A probable reason is that Balahur's method relies on noun phrases, which may mis-identify some noise noun phrases as aspects, while our approach performs hierarchical classification based on the hierarchy, which can leverage the prior knowledge encoded in the hierarchy to filter out the noise and obtain accurate aspects.

Moreover, we evaluated the effectiveness of our approach on implicit aspect identification. The 70 implicit aspect questions in our question corpus were used here. The method proposed by Su et al. (2008) was reimplemented as the baseline. It identifies implicit aspects by mutual clustering, and it was

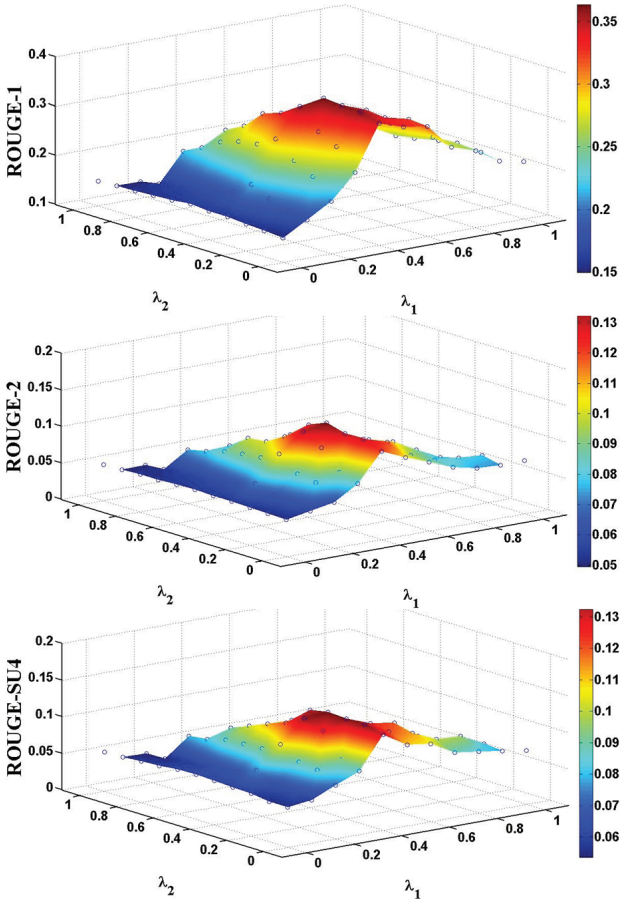


Figure 3: Evaluations on multiple optimization criteria in terms of *ROUGE-1*, *ROUGE-2*, and *ROUGE-SU4*, respectively.

evaluated in Yu et al. (2011). As shown in Table 4, our approach significantly outperforms Su's method by over 9.1% in terms of average $F_1$-measure. The results show that the hierarchy can help to identify implicit aspects by exploiting the underlying associations among sentiment terms and aspects.

| *Methods* | ROUGE1 | ROUGE2 | ROUGE-SU4 |
|---|---|---|---|
| Our method | **0.364*** | **0.137*** | **0.138*** |
| Li's method | 0.127 | 0.043 | 0.049 |
| Lloret's method | 0.149 | 0.058 | 0.065 |

Table 5: Performance of answer generation. T-Test, p-values$<$0.05.

### 4.3 Evaluations on Answer Generation

#### 4.3.1 Comparisons to the State-of-the-Arts

We compared our multi-criteria optimization approach against two state-of-the-arts methods: a) the

method presented in Li et al. (2009), which selects some retrieved sentences to generate the answers based on a graph-based algorithm; b) the method proposed by Lloret et al. (2011) that forms the answers by re-ranking the retrieved sentences.

As shown in Table 5, our approach outperforms Li's method and Lloret's method by the significant absolute gains of over 23.7%, and 21.5% respectively, in terms of average *ROUGE-1*. It improves the performance over these two methods in terms of average *ROUGE-2* by the absolute gains of over 9.41% and 7.87%, respectively; and in terms of *ROUGE-SU4* by the absolute gains of over 8.86% and 7.31%, respectively. By analyzing the results, we find that the improvements come from the use of the hierarchical organization and the answer generation algorithm which exploits multiple criteria, especially the parent-child relation among aspects. In addition, our approach can generate the answers by following the general-to-specific logic, while Li's and Lloret's methods fail to do so due to their ignorance of parent-child relations among aspects.

#### 4.3.2 Evaluations on the Effectiveness of Multiple Criteria

We further evaluated the effectiveness of each optimization criterion by tuning the trade-off parameters (i.e. $\lambda_1$, $\lambda_2$, and $\lambda_3$). We fixed $\lambda_1$ as a constant in $[0, 1]$ with 0.1 as an interval, and updated $\lambda_2$ from 0 to $1 - \lambda_1$, $\lambda_3 = 1 - \lambda_1 - \lambda_2$, correspondingly. The performance change is shown in Figure 3 in terms of *ROUGE-1*, *ROUGE-2*, and *ROUGE-SU4*, respectively. The best performance is achieved at $\lambda_1 = 0.4$, $\lambda_2 = 0.3$, $\lambda_3 = 0.3$. We observe the performance drops dramatically when any parameter (i.e. $\lambda_1$, $\lambda_2$, $\lambda_3$) is close to 0 (i.e. remove any of the corresponding criterion). Thus, we can conclude that all the criteria are useful in answer generation. We also find that the performance change is sharp when $\lambda_1$ changes. This indicates that the salience criterion is crucial for answer generation.

Table 6 shows the exemplar answers generated by our approach. Each answer first gives the statistic of positive and negative reviews. This helps user to quickly get an overview of public opinions. The summary of relevant review sentences is then presented in the answer. The answer diversely comments the asked aspect and all its available sub-aspects following the general-to-specific logic. Moreover, we feel that the answers are informative and readable.

## 5 Related Works

In this section, we review existing works related to the four components of our approach, including organization of reviews, question analysis, answer fragment retrieval, and answer generation.

For organization of reviews, Carenini et al. (2006) proposed to organize the reviews by a hand-crafted taxonomy, which was not scalable. Yu et al. (2011) exploited the domain knowledge and consumer reviews to automatically generate a hierarchy for organizing consumer reviews.

Question analysis often has to distinguish the opinion question from the factual one, and find the key points asked in the questions, such as the product aspect and product name. For example, Yu et al. (2003) proposed to separate opinions from facts at both document and sentence level, and determine the polarity on the opinionated sentences in the answer documents. Similarly, Somasundaran et al. (2007) utilized a SVM classifier to recognize opinionated sentences. The paper argued that the subjective types (i.e. sentiment and arguing) can improve the performance of opinion-QA. Later, Ku et al. (2007) proposed a two-layered classifier for question analysis, and retrieved the answer-fragments by keyword matching. In particular, they first identified the opinion questions, and classified them into six predefined question types, including holder, target, attitude, reason, majority, and yes/no. These question types and corresponding polarity on the questions were used to filter non-relevant sentences in the answer fragments. $F_1$-measure was employed as the evaluation metric.

For the topic of answer generation in opinion-QA, Li et al. (2009) formulated it as a sentence ranking task. They argued that the answers should be simultaneously relevant to topics and opinions asked in the questions. They thus designed the graph-based methods (i.e. PageRank and HITS) to select some high-ranked sentences to form answers. They first built a graph on the retrieved sentences, with each sentence as the node, and the similarity (i.e. Cosine similarity) between each sentences pair as the

| |
|---|
| *Question 1: What reasons do people give for preferring iPhone 3gs?* |
| *There are 9,928 opinionated reviews about product "iphone 3gs", with 5,717 positive and 4,221 negative reviews.*<br>*This phone is amazing and I would recommend it to anyone. It looks funky and cool. It is worth the money. It's great organiser, simple easy to use software. It is super fast, excellent connection via wifi or 3G. It is able to instantly access email. It's amazing and has so many free apps. The design is so simple and global. The hardware is good and reliable. The camera is a good and colors are vibrant. The touch screen is user friendly and the aesthetics are top notch. Battery is charged quickly, and power save right after stop using.* |
| *Question 2: Does anyone think it is expensive to get a iPhone 3GS?* |
| *Yes.*<br>*There are 2,645 opinionated reviews on aspect "price" about product "iphone 3gs", with 889 positive and 1,756 negative reviews.*<br>*Throw the costly phone, apple only knows to sell stupid stuff expensively. Don't fool yourself with iPhone 3gs, believing that it costs much by Apple luxurious advertising. Apple is so greedy and it just wants to earn easy & fast money by selling its techless product expensively. The phone will charge once you insert any sim card. iPhone 3gs is high-priced due to the capacitive and Apple license. You need to pay every application at the end it costs too much. The network provider will make up some of the cost of the phone on your call charges.* |

Table 6: Sample answers of our approach.

weight of the corresponding edge. Given a question, its similarity to each sentence in the graph was computed. Such similarity was viewed as the relevant score to the corresponding sentence. The sentences then were ranked based on three metric, i.e. relevant score to the query, similarity score obtained from the graph algorithm over sentences, and degree of opinion matching to the query. Respectively, Lloret et al. (2011) proposed to form answers by re-ranking the retrieved sentences based on the metric of word frequency, non-redundancy and the number of noun phrases. Their method includes three components, including information retrieval, opinion mining and text summarization. Evaluations were conducted on the TAC 2008 Opinion Summarization track. Afterwards, Moghaddam et al. (2011) developed a system called *AQA* to generate answers for questions about products (i.e. opinion QA on products). It classifies the questions into five types, including target, attitude, reason, majority and yes/no. As compared to Ku et al. (2007), the question types of holder and majority are not included. They argued that product questions were seldom asked for the holders, since the holders (i.e. reviewers) were commonly shown in the reviews. Also, product questions mainly asked for majority opinions, and majority type was thus not considered. The *AQA* system includes five components, including question analysis, question expansion, high quality review retrieval, subjective sentence extraction, and answer grouping. The answers are generated by aggregating opinions in the retrieved fragments.

## 6 Conclusions and Future Works

In this paper, we have developed a new product opinion-QA framework, which exploits the hierarchical organization of consumer reviews on products. With the help of the hierarchical organization, our framework can accurately identify the aspects asked in the questions and also discover their sub-aspects. We have further formulated the answer generation from retrieved review sentences as a multi-criteria optimization problem. The multiple criteria used include answer salience, diversity, and coherence. The parent-child relations between the aspects are incorporated into the approach to ensure that the answers follow the general-to-specific logic. The proposed framework has been evaluated on 11 popular products in four domains using 220 questions on the products. Significant performance improvements were obtained. In the future, we will explore the more sophisticated NLP features to improve the proposed framework. This will be done by incorporating more NLP features in salience and coherence weights estimation.

## Acknowledgments

This work is supported in part by NUS-Tsinghua Extreme Search (NExT) project under the grant number: R-252-300-001-490. We give warm thanks to the project and anonymous reviewers for their comments.

## References


E. Agichtein, C. Castillo, and D. Donato. Finding High-Quality Content in Social Media. *WSDM*, 2008.

A. Balahur, E. Boldrini, O. Ferrandez, A. Montoyo, M. Palomar, and R. Munoz. The DLSIUAES Team's Participation in the TAC 2008 Tracks. *TAC*, 2008.

C. Cardie, J. Wiebe, T. Wilson, and D. Litman. Combining Low-level and Summary Representations of Opinions for Multi-Perspective Question Answering. *AAAI*, 2003.

G. Carenini, R. Ng, and E. Zwart. Multi-document Summarization of Evaluative Text. *ACL*, 2006.

P. Cimiano. Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. *Springer-Verlag New York, Inc. Secaucus, NJ, USA*, 2006.

K. Crammer, O. Dekel, J. Keshet, S.S. Shwartz, and Y. Singer. Online Passive Aggressive Algorithms. *Journal of Machine Learning Research*, 2006.

P. Deshpande, R. Barzilay, and D.R. Karger. Randomized Decoding for Selection-and-Ordering Problems. *NAACL*, 2007.

G. Erkan and D.R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *AAAI*, 2004.

T. Givon. Syntax: A functional-typological Introduction. *Benjamins Pub*, 1990.

J. He and D. Dai. Summarization of Yes/No Questions Using a Feature Function Model. *JMLR*, 2011.

P. Jiang, H. Fu, C. Zhang, and Z. Niu. A Framework for Opinion Question Answering. *IMS*, 2010.

H.D. Kim, D.H. Park, V.G.V. Vydiswaran, and C.X. Zhai. Opinion Summarization using Entity Features and Probabilistic Sentence Coherence Optimization: UIUC at TAC 2008 Opinion Summarization Pilot. *TAC*, 2008.

D. Koller and M. Sahami. Hierarchically Classifying Documents Using Very Few Words. *ICML*, 1997.

L.W. Ku, Y.T. Liang, and H.H. Chen. Question Analysis and Answer Passage Retrieval for Opinion Question Answering Systems. *International Journal of Computational Linguistics & Chinese Language Processing*, 2007.

M. Lapata. Probabilistic Text Structuring: Experiments with Sentence Ordering. *ACL*, 2003.

M. Lapata. Automatic Evaluation of Information Ordering: Kendallś Tau. *Computational Linguistics*, 2006.

F. Li, Y. Tang, M. Huang, and X. Zhu. Answering Opinion Questions with Random Walks on Graphs. *ACL/AFNLP*, 2009.

C.Y. Lin and E.Hovy. Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. *HLT-NAACL*, 2003.

Y. Liu, X. Huang, A. An, and X. Yu. Modeling and Predicting the Helpfulness of Online Reviews. *ICDM*, 2008.

E. Lloret, A. Balahur, M. Palomar, and A. Montoyo. Towards a Unified Approach for Opinion Question Answering and Summarization. *ACL-HLT*, 2011.

H. Nishikawa, T. Hasegawa, Y. Matsuo, and G. Kikui. Opinion Summarization with Integer Linear Programming Formulation for Sentence Extraction and Ordering. *COLING*, 2010.

C.D. Mizil and G. Kossinets and J. Kleinberg and L. Lee. How Opinions are Received by Online Communities: A Case Study on Amazon.com Helpfulness Votes. *WWW*, 2009.

S. Moghaddam and M. Ester. AQA: Aspect-based Opinion Question Answering. *IEEE-ICDMW*, 2011.

Y. Ouyang, W. Li, and Q. Lu. An Integrated Multi-document Summarization Approach based on Word Hierarchical Representation. *ACL-IJCNLP*, 2009.

A. Schrijver. Theory of Linear and Integer Programming. *John Wiley & Sons*, 1998.

D.W. Scott. Multivariate Density Estimation: Theory, Practice, and Visualization. *John Wiley & Sons, Inc.*, 1992.

C. Silla and A. Freitas. A Survey of Hierarchical Classification Across Different Application Domains. *Data Mining and Knowledge Discovery*, 2011.

S. Somasundaran, T. Wilson, J. Wiebe and V. Stoyanov. QA with Attitude: Exploiting Opinion Type Analysis for Improving Question Answering in Online Discussions and the News. *ICWSM*, 2007.

V. Stoyanov, C. Cardie and J. Wiebe. Multi-Perspective Question Answering Using the OpQA Corpus. *EMNLP*, 2005.

Q. Su, X. Xu, H. Guo, X. Wu, X. Zhang, B. Swen, and Z. Su. Hidden Sentiment Association in Chinese Web Opinion Mining. *WWW*, 2008.

T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. *HLT/EMNLP*, 2005.

J. Yu, Z.J. Zha, M. Wang, K. Wang and T.S. Chua. Domain-Assisted Product Aspect Hierarchy Generation: Towards Hierarchical Organization of Unstructured Consumer Reviews. *EMNLP*, 2011.

J. Yu, Z.J. Zha, M. Wang, and T.S. Chua. Hierarchical Organization of Unstructured Consumer Reviews. *WWW*, 2011.

J. Yu, Z.J. Zha, M. Wang and T.S. Chua. Aspect Ranking: Identifying Important Product Aspects from Online Consumer Reviews. *ACL*, 2011.

H. Yu and V. Hatzivassiloglou. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. *EMNLP*, 2003.