Domain-Assisted Product Aspect Hierarchy Generation: Towards Hierarchical Organization of Unstructured Consumer Reviews

Jianxing Yu[1], **Zheng-Jun Zha**[1], **Meng Wang**[1], **Kai Wang**[2], **Tat-Seng Chua**[1]
[1]School of Computing, National University of Singapore
[2]Institute for Infocomm Research, Singapore
{jianxing, zhazj, wangm, chuats}@comp.nus.edu.sg kwang@i2r.a-star.edu.sg

Abstract

This paper presents a domain-assisted approach to organize various aspects of a product into a hierarchy by integrating domain knowledge (e.g., the product specifications), as well as consumer reviews. Based on the derived hierarchy, we generate a hierarchical organization of consumer reviews on various product aspects and aggregate consumer opinions on these aspects. With such organization, user can easily grasp the overview of consumer reviews. Furthermore, we apply the hierarchy to the task of implicit aspect identification which aims to infer implicit aspects of the reviews that do not explicitly express those aspects but actually comment on them. The experimental results on 11 popular products in four domains demonstrate the effectiveness of our approach.

1 Introduction

With the rapidly expanding e-commerce, most retail Web sites encourage consumers to write reviews to express their opinions on various aspects of products. Huge collections of consumer reviews are now available on the Web. These reviews have become an important resource for both consumers and firms. Consumers commonly seek quality information from online consumer reviews prior to purchasing a product, while many firms use online reviews as an important resource in their product development, marketing, and consumer relationship management. However, the reviews are disorganized, leading to the difficulty in information navigation and knowledge acquisition. It is impractical for user to grasp the overview of consumer reviews and opinions on various aspects of a product from such enormous reviews. Among hundreds of product aspects, it is also inefficient for user to browse consumer reviews and opinions on a specific aspect. Thus, there is a compelling need to organize consumer reviews, so as to transform the reviews into a useful knowledge structure. Since the hierarchy can improve information representation and accessibility (Cimiano, 2006), we propose to organize the aspects of a product into a hierarchy and generate a hierarchical organization of consumer reviews accordingly.

Towards automatically deriving an aspect hierarchy from the reviews, we could refer to traditional hierarchy generation methods in ontology learning, which first identify concepts from the text, then determine the parent-child relations between these concepts using either pattern-based or clustering-based methods (Murthy et al., 2010). However, pattern-based methods usually suffer from inconsistency of parent-child relationships among the concepts, while clustering-based methods often result in low accuracy. Thus, by directly utilizing these methods to generate an aspect hierarchy from consumer reviews, the resulting hierarchy is usually inaccurate, leading to unsatisfactory review organization. On the other hand, domain knowledge of products is now available on the Web. For example, there are more than 248,474 product specifications in the product selling Web site CNet.com (Beckham, 2005). These product specifications cover some product aspects and provide coarse-grained parent-child relations among these aspects. Such domain knowledge is useful to help organize the product as-

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 140–150,
Edinburgh, Scotland, UK, July 27–31, 2011. ©2011 Association for Computational Linguistics

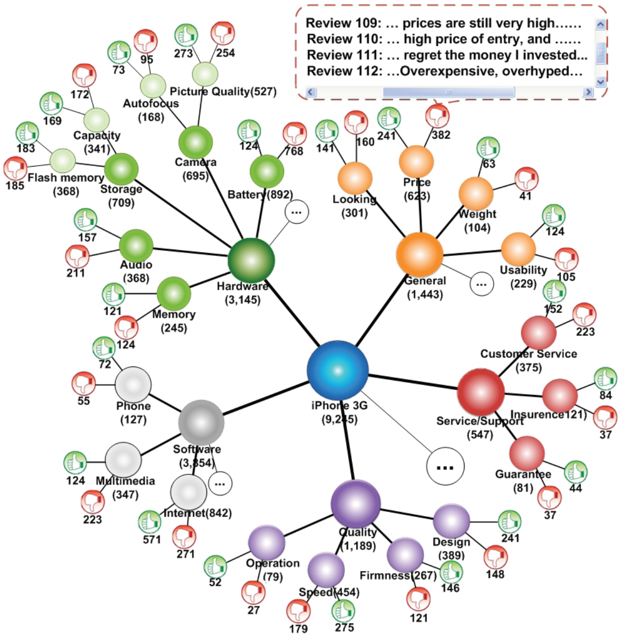

Review 109: ... prices are still very high......
Review 110: ... high price of entry, and ......
Review 111: ... regret the money I invested...
Review 112: ...Overexpensive, overhyped...

Figure 1: Sample hierarchical organization for iPhone 3G

pects into a hierarchy. However, the initial hierarchy obtained from domain knowledge usually cannot fit the review data well. For example, the initial hierarchy is usually too coarse and may not cover the specific aspects commented in the reviews, while some aspects in the hierarchy may not be of interests to users in the reviews.

Motivated by the above observations, we propose in this paper to organize the product aspects into a hierarchy by simultaneously exploiting the domain knowledge (e.g., the product specification) and consumer reviews. With derived aspect hierarchy, we generate a hierarchical organization of consumer reviews on various aspects and aggregate consumer opinions on these aspects. Figure 1 illustrates a sample of hierarchical review organization for the product "*iPhone 3G*". With such organization, users can easily grasp the overview of product aspects as well as conveniently navigate the consumer reviews and opinions on any aspect. For example, users can find that 623 reviews, out of 9,245 reviews, are about the aspect "*price*", with 241 positive and 382 negative reviews.

Given a collection of consumer reviews on a specific product, we first automatically acquire an initial aspect hierarchy from domain knowledge and identify the aspects from the reviews. Based on the initial hierarchy, we develop a multi-criteria optimization approach to construct an aspect hierarchy to contain all the identified aspects. Our approach incrementally inserts the aspects into the initial hierarchy based on inter-aspect semantic distance, a metric used to measure the semantic relation among aspects. In order to derive reliable semantic distance, we propose to leverage external hierarchies, sampled from WordNet and Open Directory Project, to assist semantic distance learning. With resultant aspect hierarchy, the consumer reviews are then organized to their corresponding aspect nodes in the hierarchy. We then perform sentiment classification to determine consumer opinions on these aspects. Furthermore, we apply the hierarchy to the task of implicit aspect identification. This task aims to infer implicit aspects of the reviews that do not explicitly express those aspects but actually comment on them. For example, the implicit aspect of the review "It is so expensive" is "price." Most existing aspect identification approaches rely on the appearance of aspect terms, and thus are not able to handle implicit aspect problem. Based on our aspect hierarchy, we can infer the implicit aspects by clustering the reviews into their corresponding aspect nodes in the hierarchy. We conduct experiments on 11 popular products in four domains. More details of the corpus are discussed in Section 4. The experimental results demonstrate the effectiveness of our approach.

The main contributions of this work can be summarized as follows:

1) We propose to hierarchically organize consumer reviews according to an aspect hierarchy, so as to transfer the reviews into a useful knowledge structure.

2) We develop a domain-assisted approach to generate an aspect hierarchy by integrating domain knowledge and consumer reviews. In order to derive reliable semantic distance between aspects, we propose to leverage external hierarchies to assist semantic distance learning.

3) We apply the aspect hierarchy to the task of implicit aspect identification, and achieve satisfactory performance.

The rest of this paper is organized as follows. Our approach is elaborated in Section 2 and applied to implicit aspect identification in Section 3. Section 4 presents the evaluations, while Section 5 reviews related work. Finally, Section 6 concludes this paper with future works.

2 Approach

Our approach consists of four components, including initial hierarchy acquisition, aspect identification, semantic distance learning, and aspect hierarchy generation. Next, we first define some preliminary and notations and then elaborate these components.

2.1 Preliminary and Notations

Preliminary 1. An aspect hierarchy is defined as a tree that consists of a set of unique aspects $\mathcal{A}$ and a set of parent-child relations $\mathcal{R}$ between these aspects.

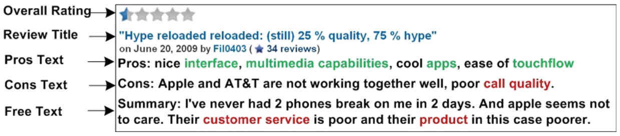
Given the consumer reviews of a product, let $\mathcal{A} = \{a_1, \cdots, a_k\}$ denotes the product aspects commented in the reviews. $\mathcal{H}^0(\mathcal{A}^0, \mathcal{R}^0)$ denotes the initial hierarchy derived from domain knowledge. It contains a set of aspects $\mathcal{A}^0$ and relations $\mathcal{R}^0$. Our task is to construct an aspect hierarchy $\mathcal{H}(\mathcal{A}, \mathcal{R})$, to cover all the aspects in $\mathcal{A}$ and their parent-child relations $\mathcal{R}$, so that the consumer reviews are hierarchically organized. Note that $\mathcal{H}^0$ can be empty.

2.2 Initial Hierarchy Acquisition

As aforementioned, product specifications on product selling websites cover some product aspects and coarse-grained parent-child relations among these aspects. Such domain knowledge is useful to help organize aspects into a hierarchy. We here employ the approach proposed by Ye and Chua (2006) to automatically acquire an initial aspect hierarchy from the product specifications. The method first identifies the Web page region covering product descriptions and removes the irrelevant contents from the Web page. It then parses the region containing the product information to identify the aspects as well as their structure. Based on the aspects and their structure, it generates an aspect hierarchy.

2.3 Aspect Identification

To identify aspects in consumer reviews, we first parse each review using the Stanford parser [1]. Since the aspects in consumer reviews are usually noun or noun phrases (Liu, 2009), we extract the noun phrases (*NP*) from the parse tree as aspect candidates. While these candidates may contain much noise, we leverage *Pros* and *Cons* reviews (see Figure 2), which are prevalent in forum Web sites, to assist identify aspects from the candidates. It has been shown that simply extracting the frequent noun terms from the *Pros* and *Cons* reviews can get high accurate aspect terms (Liu el al., 2005). Thus, we extract the frequent noun terms from *Pros* and *Cons* reviews as features, then train a one-class *SVM* (Manevitz et al., 2002) to identify aspects from the candidates. While the obtained aspects may contain some synonym terms, such as “earphone” and “headphone”, we further perform synonym clustering to get unique aspects. Specifically, we first expand each aspect term with its synonym terms obtained from the synonym terms Web site [2], then cluster them to obtain unique aspects based on unigram feature.



"Hype reloaded reloaded: (still) 25 % quality, 75 % hype"
on June 20, 2009 by Fil0403 (★ 34 reviews)
Pros: nice interface, multimedia capabilities, cool apps, ease of touchflow
Cons: Apple and AT&T are not working together well, poor call quality.
Summary: I've never had 2 phones break on me in 2 days. And apple seems not to care. Their customer service is poor and their product in this case poorer.

Figure 2: Sample Pros and Cons reviews

2.4 Semantic Distance Learning

Our aspect hierarchy generation approach is essentially based on the semantic relations among aspects. We here define a metric, *Semantic Distance*, $d(a_x, a_y)$, to quantitatively measure the semantic relation between aspects a_x and a_y. We formulate $d(a_x, a_y)$ as the weighted sum of some underlying features,

$$d(a_x, a_y) = \sum\nolimits_j w_j f_j(a_x, a_y), \qquad (1)$$

where w_j is the weight for j-th feature function $f_j(\cdot)$.

Next, we first introduce the linguistic features used in our work and then present the semantic distance learning algorithm that aims to find the optimal weights in Eq.(1).

[1] http://nlp.stanford.edu/software/lex-parser.shtml

[2] http://thesaurus.com

2.4.1 Linguistic Features

Given two aspects a_x and a_y, a feature is defined as a function generating a numeric score $f(a_x, a_y)$ or a vector of scores. The features include *Contextual*, *Co-occurrence*, *Syntactic*, *Pattern* and *Lexical* features (Yang and Callan, 2009). These features are generated based on auxiliary documents collected from Web.

Specifically, we issue each aspect term and aspect term pair as queries to Google and Wikipedia, respectively, and collect the top 100 returned documents of each query. We then split the documents into sentences. Based on these documents and sentences, the features are generated as follows.

Contextual features. For each aspect, we collect the documents containing the aspect as context to build a unigram language model without smoothing. Given two aspects, the KL-divergence between their language models is computed as the *Global-Context* feature between them. Similarly, we collect the left two and right two words surrounding each aspect as context and build a unigram language model. The KL-divergence between the language models of two aspects is defined as the *Local-Context* feature.

Co-occurrence features. We measure the co-occurrence of two aspects by Pointwise Mutual Information (PMI): *PMI*(a_x,a_y)=log(*Count*(a_x,a_y)/ *Count*(a_x) *Count*(a_y)), where *Count*($\cdot$) stands for the number of documents or sentences containing the aspect(s), or the number of Google document hits for the aspect(s). Based on different definitions of *Count*($\cdot$), we define the features of *Document PMI*, *Sentence PMI*, and *Google PMI*, respectively.

Syntactic features. We parse the sentences that contain each aspect pair into syntactic trees via the Stanford Parser. The *Syntactic-path* feature is defined as the average length of the shortest syntactic path between the aspect pair in the tree. In addition, for each aspect, we collect a set of sentences containing it, and label the semantic role of the sentences via ASSERT parser [3]. Given two aspects, the number of the Subject terms overlaps between their sentence sets is computed as the *Subject Overlap* feature. Similarly, for other semantic roles, such as objects, modifiers, and verbs, we define the features of *Object Overlap*, *Modifier Overlap*, and *Verb Overlap*, respectively.

Pattern features. 46 patterns are used in our work, including 6 patterns indicating the hypernym relations of two aspects (Hearst, 1992), and 40 patterns measuring the part-of relations of two aspects (Girju et al., 2006). These pattern features are asymmetric, and they take the parent-child relations among the aspects into consideration. All the patterns are listed in Appendix A (submitted as supplementary material). Based on these patterns, a 46-dimensional score vector is obtained for each aspect pair. A score is 1 if two aspects match a pattern, and 0 otherwise.

Lexical features. We take the word length difference between two aspects, as *Length Difference* feature. In addition, we issue the query “define:aspect” to Google, and collect the definition of each aspect. We then count the word overlaps between the definitions of two aspects, as *Definition Overlap* feature.

2.4.2 Semantic Distance Learning

This section elaborates the learning algorithm that optimizes the semantic distance metric, i.e., the weighting parameters in Eq.(1). Typically, we can utilize the initial hierarchy as training data. The ground-truth distance between two aspects $d^{\mathcal{G}}(a_x, a_y)$ is generated by summing up all the edge distances along the shortest path between a_x and a_y, where every edge weight is assumed as 1. The distance metric is then obtained by solving the following optimization problem,

$$\arg\min_{w_j|_{j=1}^{m}} \sum_{\substack{a_x,a_y\in\mathcal{A}^0\\ x<y}} (d^{\mathcal{G}}(a_x,a_y) - \sum_{j=1}^{m} w_j f_j(a_x,a_y))^2 + \eta\cdot\sum_{j=1}^{m} w_j^2, \tag{2}$$

where m is the dimension of linguistic feature, η is a tradeoff parameter. Eq.(2) can be rewrote to its matrix form as,

$$\arg\min_{\mathbf{w}} \left\|\mathbf{d} - \mathbf{f}^T\mathbf{w}\right\|^2 + \eta\cdot\|\mathbf{w}\|^2, \tag{3}$$

where vector **d** contains the ground-truth distance of all the aspect pairs. Each element corresponds to the distance of certain aspect pair, and **f** is the corresponding feature vector. The optimal solution of **w** is given as

$$\mathbf{w}^* = (\mathbf{f}^T\mathbf{f} + \eta\cdot\mathbf{I})^{-1}(\mathbf{f}^T\mathbf{d}) \tag{4}$$

[3] http://cemantix.org/assert.html

where **I** is the identity metric.

The above learning algorithm can perform well when sufficient training data (i.e., aspect (term) pairs) is available. However, the initial hierarchy is usually too coarse and thus cannot provide sufficient information. On the other hand, abundant handcrafted hierarchies are available on the Web, such as WordNet and Open Directory Project (ODP). We here propose to leverage these external hierarchies to assist semantic distance learning. A distance metric $\mathbf{w}_0$ is learned from the external hierarchies using the above algorithm. Since $\mathbf{w}_0$ might be biased to the characteristics of the external hierarchies, directly using $\mathbf{w}_0$ in our task may not perform well. Alternatively, we use $\mathbf{w}_0$ as prior knowledge to assist learning the optimal distance metric **w** from the initial hierarchy. The learning problem is formulated as follows,

$$\arg\min_{\mathbf{w}} \left\|\mathbf{d} - \mathbf{f}^T\mathbf{w}\right\|^2 + \eta \cdot \|\mathbf{w}\|^2 + \gamma \cdot \|\mathbf{w} - \mathbf{w}_0\|^2, \tag{5}$$

where η and γ are tradeoff parameters.

The optimal solution of **w** can be obtained as

$$\mathbf{w}^* = (\mathbf{f}^T\mathbf{f} + (\eta + \gamma) \cdot \mathbf{I})^{-1}(\mathbf{f}^T\mathbf{d} + \gamma \cdot \mathbf{w}_0). \tag{6}$$

As a result, we can compute the semantic distance between each two aspects according to Eq.(1).

2.5 Aspect Hierarchy Generation

Given the aspects $\mathcal{A} = \{a_1, \cdots, a_k\}$ identified from reviews and the initial hierarchy $\mathcal{H}^0(\mathcal{A}^0, \mathcal{R}^0)$ obtained from domain knowledge, our task is to construct an aspect hierarchy to contain all the aspects in $\mathcal{A}$. Inspired by Yang and Callan (2009), we adopt a multi-criteria optimization approach to incrementally insert the aspects into appropriate positions in the hierarchy based on multiple criteria.

Before going to the details, we first introduce an information function to measure the amount of information carried in a hierarchy. An information function $Info(\mathcal{H})$ is defined as the sum of the semantic distances of all the aspect pairs in the hierarchy (Yang and Callan, 2009).

$$Info(\mathcal{H}(\mathcal{A}, \mathcal{R})) = \sum_{x<y; a_x, a_y \in \mathcal{A}} d(a_x, a_y). \tag{7}$$

Based on this information function, we then introduce the following three criteria for aspect insertion: *minimum Hierarchy Evolution*, *minimum Hierarchy Discrepancy* and *minimum Semantic Inconsistency*.

Hierarchy Evolution is designed to monitor the structure evolution of a hierarchy. The hierarchy is incrementally hosting more aspects until all the aspects are allocated. The insertion of a new aspect a into different positions in the current hierarchy $\mathcal{H}^{(i)}$ leads to different new hierarchies. Among these new hierarchies, we here assume that the optimal one $\mathcal{H}^{(i+1)}$ should introduce the least changes of information to $\mathcal{H}^{(i)}$.

$$\hat{\mathcal{H}}^{(i+1)} = \arg\min_{\mathcal{H}^{(i+1)}} \Delta Info(\mathcal{H}^{(i+1)} - \mathcal{H}^{(i)}). \tag{8}$$

By plugging in Eq.(7) and using *least square* to measure the information changes, we have,

$$\begin{aligned} obj_1 = \arg\min_{\mathcal{H}^{(i+1)}} (&\textstyle\sum_{x<y; a_x, a_y \in \mathcal{A}_i \cup \{a\}} d(a_x, a_y) \\ &- \textstyle\sum_{x<y; a_x, a_y \in \mathcal{A}_i} d(a_x, a_y))^2, \end{aligned} \tag{9}$$

Hierarchy Discrepancy is used to measure the global changes of the structure. We assume a good hierarchy should bring the least changes to the initial hierarchy,

$$\hat{\mathcal{H}}^{(i+1)} = \arg\min_{\mathcal{H}^{(i+1)}} \frac{\Delta Info(\mathcal{H}^{(i+1)} - \mathcal{H}^{(0)})}{i+1}. \tag{10}$$

We then get,

$$\begin{aligned} obj_2 = \arg\min_{\mathcal{H}^{(i+1)}} \tfrac{1}{i+1}(&\textstyle\sum_{x<y; a_x, a_y \in \mathcal{A}_i \cup \{a\}} d(a_x, a_y) \\ &- \textstyle\sum_{x<y; a_x, a_y \in \mathcal{A}_0} d(a_x, a_y))^2. \end{aligned} \tag{11}$$

Semantic Inconsistency is introduced to quantify the inconsistency between the semantic distance estimated via the hierarchy and that computed from the feature functions. We assume that a good hierarchy should precisely reflect the semantic distance between aspects. For two aspects, their semantic distance reflected by the hierarchy is computed as the sum of adjacent distances along the shortest path between them,

$$d^{\mathcal{H}}(a_x, a_y) = \textstyle\sum_{p<q; (a_p, a_q) \in SP(a_x, a_y)} d(a_p, a_q), \tag{12}$$

where $SP(a_x, a_y)$ is the shortest path between the aspects (a_x, a_y), (a_p, a_q) are the adjacent nodes along the path.

We then define the following criteria to find the hierarchy with minimum semantic inconsistency,

$$obj_3 = \arg\min_{\mathcal{H}^{(i+1)}} \sum_{x<y; a_x, a_y \in \mathcal{A}_i \cup \{a\};} (d^{\mathcal{H}}(a_x, a_y) - d(a_x, a_y))^2, \tag{13}$$

where $d(a_x, a_y)$ is the distance computed based on the feature functions in Section 2.4.

Through integrating the above criteria, the multi-criteria optimization framework is formulated as,

$$\begin{aligned} &obj = \arg\min_{\mathcal{H}^{(i+1)}} (\lambda_1 \cdot obj_1 + \lambda_2 \cdot obj_2 + \lambda_3 \cdot obj_3) \\ &\lambda_1 + \lambda_2 + \lambda_3 = 1; \;\; 0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1. \end{aligned} \tag{14}$$

where λ_1, λ_2, λ_3 are the tradeoff parameters.

To summarize, our aspect hierarchy generation process starts from an initial hierarchy and inserts the aspects into it one-by-one until all the aspects are allocated. Each aspect is inserted to the optimal position found by Eq.(14). It is worth noting that the insertion order may influence the result. To avoid such influence, we select the aspect with the least objective function value in Eq.(14) to insert. Based on resultant hierarchy, the consumer reviews are then organized to their corresponding aspect nodes in the hierarchy. We further prune out the nodes without reviews from the hierarchy.

Moreover, we perform sentiment classification to determine consumer opinions on various aspects. In particular, we train a *SVM* sentiment classifier based on the *Pros* and *Cons* reviews described in Section 2.3. We collect sentiment terms in the reviews as features and represent reviews as feature vectors using Boolean weighting. Note that we define sentiment terms as those appear in the sentiment lexicon provided by *MPQA* project (Wilson et al., 2005).

3 Implicit Aspect Identification

In this section, we apply the aspect hierarchy to the task of implicit aspect identification. This task aims to infer the aspects of reviews that do not explicitly express those aspects but actually comment on them (Liu et al. 2005). Take the review "The phone is too large" as an example, the task is to infer its implicit aspect "size." It has been observed that the reviews commenting on a same aspect usually use some same sentiment terms (Su et al., 2008). Therefore, sentiment term is an effective feature for identifying implicit aspects. We here collect the sentiment terms as features to represent each review into a feature vector. For each aspect node in the hierarchy, we define its centroid as the average of its feature vectors, i.e., the feature vectors of all the reviews that are allocated at this node. We then calculate the cosine similarity of each implicit-aspect review to the centriods of all the aspect nodes, and allocate the review into the node with maximum similarity. As a result, the implicit aspect reviews are grouped to their related aspect nodes. In other word, their aspects are identified as the corresponding aspect nodes.

4 Evaluations

In this section, we evaluate the effectiveness of our approach on aspect identification, aspect hierarchy generation, and implicit aspect identification.

4.1 Data and Experimental Setting

The details of our product review corpus are given in Table 1. This corpus contains consumer reviews on 11 popular products in four domains. These reviews were crawled from several prevalent forum Web sites, including cnet.com, viewpoints.com, reevoo.com and gsmarena.com. All of the reviews were posted between June, 2009 and Sep 2010. The aspects of the reviews, as well as the opinions on the aspects were manually annotated. We also invited five annotators to construct the gold-standard hierarchies for the products by providing them the initial hierarchies and the aspects in reviews. The conflicts between annotators were resolved through their discussions. For semantic distance learning, we collected 50 hierarchies from WordNet and ODP, respectively. The details are shown in Table 2. We listed the topics of the hierarchies in Appendix B (submitted as supplementary material).

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 reviews corpus, # denotes the size of the reviews/sentences

Statistic	*WordNet*	*ODP*
Total # hierarchies	50	50
Total # terms	1,964	2,210
Average # depth	5.5	5.9
Total # related topics	12	16

Table 2: Statistics of the External Hierarchies

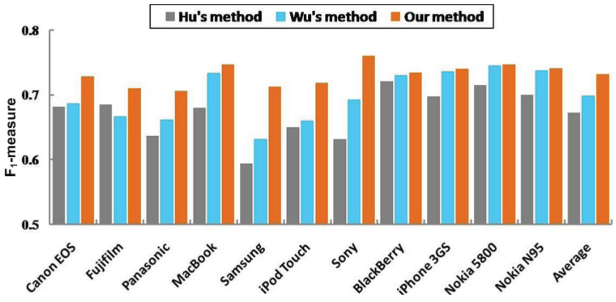


Figure 3: Evaluations on Aspect Identification. t-test, p-values<0.05

We employed F_1-measure, which is the combination of *precision* and *recall*, as the evaluation metric for all the evaluations. For the evaluation on aspect hierarchy, we defined *precision* as the percentage of correctly returned parent-child pairs out of the total returned pairs, and *recall* as the percentage of correctly returned parent-child pairs out of the total pairs in the gold standard. Throughout the experiments, we empirically set $\lambda_1 = 0.4$, $\lambda_2 = 0.3$, $\lambda_3 = 0.3$, $\eta = 0.4$ and $\gamma = 0.6$.

4.2 Evaluations on Aspect Identification

We compared our approach against two state-of-the-art methods: a) the method proposed by Hu and Liu (2004), which is based on the association rule mining, and b) the method proposed by Wu et al. (2009), which is based on the dependency parser. The results are presented in Figure 3. On average, our approach significantly outperforms Hu's and Wu's method in terms of F_1-measure by over 5.87% and 3.27%, respectively.

4.3 Evaluations on Aspect Hierarchy

4.3.1 Comparisons with the State-of-the-Arts

We compared our approach against four traditional hierarchy generation methods in the researches on ontology learning, including a) pattern-based method (Hearst, 1992) and b) clustering-based method by Shi et al. (2008), c) the method proposed by Snow et al. (2006) which was based on a probabilistic model, and d) the method proposed by Yang and Callan (2009). Since our approach and Yang's method can utilize initial hierarchy to assist hierarchy generation, we evaluated their performance with or without initial hierarchy, respectively. For the sake of fair comparison, Snow's, Yang's and our methods used the same linguistic features in Section 2.4.1.

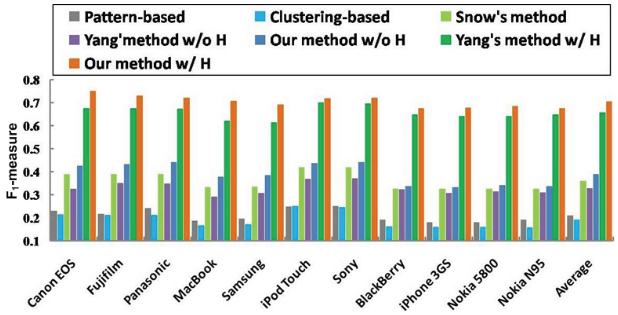


Figure 4: Evaluations on Aspect Hierarchy Generation. t-test, p-values<0.05. w/ H denotes the methods with initial hierarchy, accordingly, w/o H refers to the methods without initial hierarchy.

Figure 4 shows the performance comparisons of these five methods. We can see that our approach without using initial hierarchy outperforms the pattern-based, clustering-based, Snow's, and Yang's methods by over 17.9%, 19.8%, 2.9% and 6.1% respectively in terms of average F_1-measure. By exploiting initial hierarchy, our approach improves the performance significantly. As compared to the pattern-based, clustering-based and Snow's methods, it improves the average performance by over 49.4%, 51.2% and 34.3% respectively. Compared to Yang's method with initial hierarchy, it achieves 4.7% improvements on the average performance.

The results show that pattern-based and clustering-based methods perform poor. Pattern-based method may suffer from the problem of low coverage of patterns, especially when the patterns are manually pre-defined, while the clustering-based method (Shi et al., 2008) may sustain to the bisection clustering mechanism which can only generate a binary-tree. The results also illustrate that our approach outperforms Snow's and Yang's methods. By exploiting external hierarchies, our

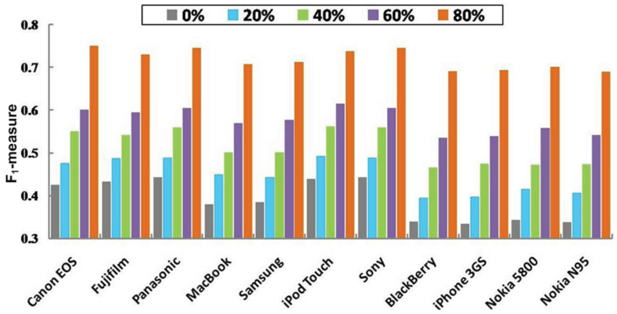


Figure 5: Evaluations on the Impact of Initial Hierarchy. t-test, p-values<0.05.

approach is able to derive reliable semantic distance between aspects and thus improve the performance.

4.3.2 Evaluations on Effectiveness of Initial Hierarchy

In this section, we show that even based on a small part of the initial hierarchy, our approach can still generate a satisfactory hierarchy. We explored different proportion of initial hierarchy, including 0%, 20%, 40%, 60% and 80% of the aspect pairs which are collected top-down from the initial hierarchy. As shown in Figure 5, the performance increases when larger proportion of the initial hierarchy is used. Thus, we can speculate that domain knowledge is valuable in aspect hierarchy generation.

4.3.3 Evaluations on Effectiveness of Optimization Criteria

We conducted a leave-one-out study to evaluate the effectiveness of each optimization criterion. In particular, we set one of the tradeoff parameters (λ_1, λ_2, λ_3) in Eq.(14) to zero, and distributed its weight to the rest parameters averagely. From Figure 6, we find that removing any optimization criterion would degrade the performance on most products. It is interesting to note that removing the third optimization criterion, i.e., minimum semantic inconsistency, slightly increases the performance on two products (ipad touch and sony MP3). The reason might be that the values of the three tradeoff parameters (empirically set in Section 4.1) are not suitable for these two products.

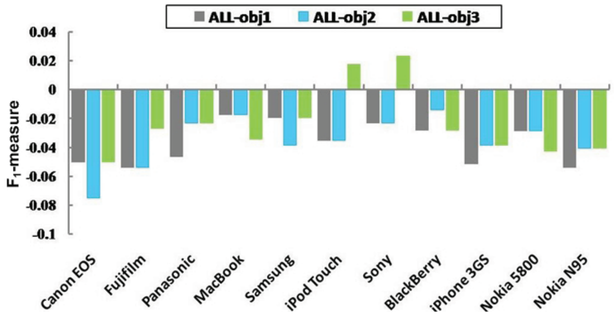


Figure 6: Evaluations of the Optimization Criteria. % of change in F_1-measure when a single criterion is removed. t-test, p-values<0.05.

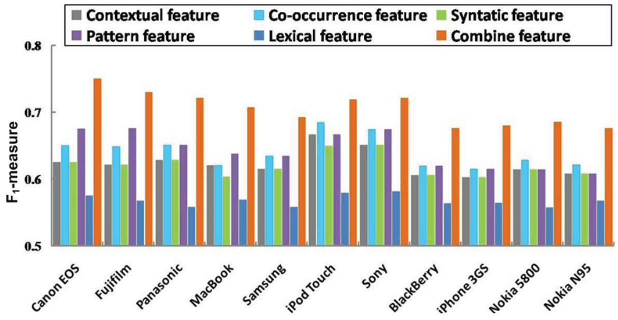


Figure 7: Evaluations on the Impact of Linguistic Features. t-test, p-values<0.05.

4.3.4 Evaluations on Semantic Distance Learning

In this section, we evaluated the impact of the features and external hierarchies in semantic distance learning. We investigated five sets of features as described in Section 2.4.1, including contextual, co-occurrence, syntactic, pattern and lexical features. From Figure 7, we observe that the co-occurrence and pattern features perform much better than contextual and syntactic features. A possible reason is that co-occurrence and pattern features are more likely to indicate parent-child aspect relationships, while contextual and syntactic features are probable to measure sibling aspect relationships. Among these features, the lexical features perform the worst. The combination of all the features achieves the best performance.

Next, we evaluated the effectiveness of external hierarchies in semantic distance learning. We compared the performance of our approach with or without the external hierarchies. From Figure 8, we find that by exploiting the external hierarchies, our ap-

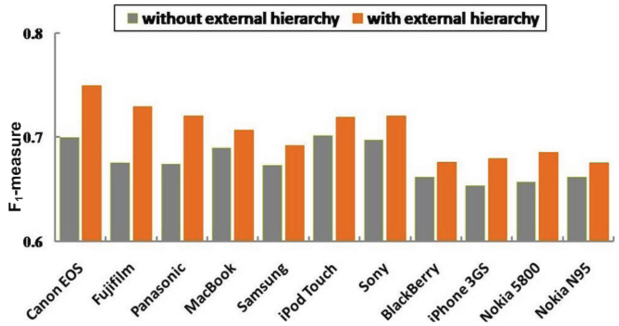


Figure 8: Evaluations on the Impact of External Hierarchy. t-test, p-values<0.05.

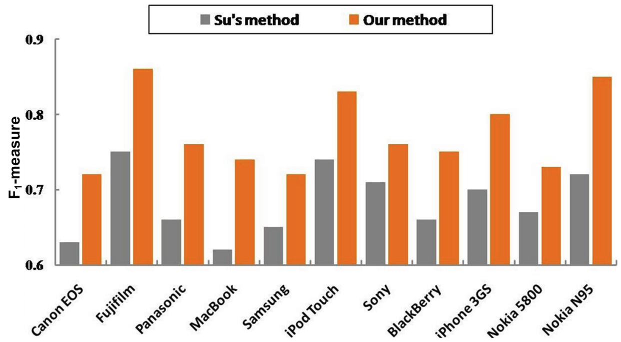


Figure 9: Evaluations on Implicit Aspects Identification. t-test, p-values<0.05

proach improves the performance significantly. The improvement is over 2.81% in terms of average F_1-measure. This implies that by using external hierarchies, our approach can obtain effective semantic distance, and thus improve the performance of aspect hierarchy generation.

Additionally, for sentiment classification, our *SVM* classifier achieves an average F_1-measure of 0.787 in the 11 products.

4.4 Evaluations on Implicit Aspect Identification

To evaluate the performance of our approach on implicit aspect identification, we collected 29,657 implicit aspect review sentences on the 11 products from the four forum Web sites introduced in Section 4.1. While most existing approaches for implicit aspect identification rely on hand-crafted rules (Liu, 2009), the method proposed in Su et al. (2008) can identify implicit aspects without hand-crafted rules based on mutual clustering. Therefore, we adopt Su's method as the baseline here. Figure 9 illustrates the performance comparison between Su's and our approach. We can see that our approach outperforms Su's method by over 9.18% in terms of average F_1-measure. This shows that our approach can identify the implicit aspects accurately by exploiting the underlying associations among the sentiment terms and each aspect in the hierarchy.

5 Related Work

Some researches treated review organization as a multi-document summarization problem, and generated a summary by selecting and ordering sentences taken from multiple reviews (Nishikawa et al., 2010). These works did not drill down to the fine-grained level to explore the opinions on the product aspects. Other researchers proposed to produce a summary covering consumer opinions on each aspect. For example, Hu and Liu (2004) focused on extracting the aspects and determining opinions on the aspects. However, their generated summary was unstructured, where the possible relationships between aspects were not recognized (Cadilhac et al., 2010). Subsequently, Carenini et al. (2006) proposed to map the aspect to a user-defined taxonomy, but the taxonomy was hand-crafted which was not scalable.

Different from the previous works, we focus on automatically generating an aspect hierarchy to hierarchically organize consumer reviews. There are some related works on ontology learning, which first identify concepts from text, and then determine parent-child relations between these concepts using either pattern-based or clustering-based methods (Murthy et al., 2010). Pattern-based methods usually defined some lexical syntactic patterns to extract the relations, while clustering-based methods mostly utilized the hierarchical clustering methods to build a hierarchy (Roy et al., 2006). Some works proposed to integrate the pattern-based and clustering-based methods in a general model, such as the probabilistic model (Snow et al., 2006) and metric-based model (Yang and Callan, 2009).

The researches on aspect identification are also related to our work. Various aspect identification methods have been proposed (Popescu et al., 2005), including supervised methods (Liu el al., 2005), and unsupervised methods (Mei et al., 2007). Different features have been investigated for this task. For example, Wu et al. (2009) identified aspects based on the features explored by dependency parser. For implicit aspect identification, some works proposed to define rules for identification (Liu el al., 2005), while others suggested to automatically generate rules via mutual clustering (Su et al., 2008). On the other hand, there are some related works on sentiment classification (Pang and Lee, 2008). These works can be categorized into four granularities: document-level, sentence-level, aspect-level and word-level sentiment classification (Liu, 2009). Existing researches have been studied unsupervised (Kim et al., 2004), supervised (Pang et al., 2002; Pang et al., 2005) and semi-supervised classification approaches (Goldberg et al., 2006; Li et al., 2009) on these four levels.

6 Conclusions and Future Works

In this paper, we have developed a domain-assisted approach to generate product aspect hierarchy by integrating domain knowledge and consumer reviews. Based on the derived hierarchy, we can generate a hierarchical organization of consumer reviews as well as consumer opinions on the aspects. With such organization, user can easily grasp the overview of consumer reviews, as well as seek consumer reviews and opinions on any specific aspect by navigating through the hierarchy. We have further applied the hierarchy to the task of implicit aspect identification. We have conducted evaluations on 11 different products in four domains. The experimental results have demonstrated the effectiveness of our approach. In the future, we will explore other linguistic features to learn the semantic distance between aspects, as well as apply our approach to other applications.

Acknowledgments

This work is supported 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 valuable comments.

References

P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan. An Exploration of Sentiment Summarization. *AAAI*, 2003.

J. Beckham. The Cnet E-commerce Data set. *Technical Reports*, 2005.

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

A. Cadilhac, F. Benamara, and N. Aussenac-Gilles. Ontolexical Resources for Feature based Opinion Mining: a Case-study. *Ontolex*, 2010.

P. Cimiano, A. Madche, S. Staab, and J. Volker. Ontology Learning. *Handbook on Ontologies, Springer*, 2004.

P. Cimiano, A. Hotho, and S. Staab. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Artificial Intelligence*, 2005.

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

S. Dasgupta and V. Ng. Mine the Easy, Classify the Hard: A Semi-supervised Approach to Automatic Sentiment Classification. *ACL*, 2009.

O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Unsupervised Named-entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*, 2005.

A. Esuli and F. Sebastiani. A Publicly Available Lexical Resource for Opinion Mining. *LREC*, 2006.

M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining Customer Opinions from Free Text. *IDA*, 2005.

R. Girju and A. Badulescu. Automatic Discovery of Part-whole Relations *Computational Linguistics*, 2006.

A. Goldberg and X. Zhu. Seeing Stars When There Aren't Many Stars: Graph-based Semi-supervised Learning for Sentiment Categorization. *ACL*, 2006.

M.A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. *Coling*, 1992.

M. Hu and B. Liu. Mining and Summarizing Customer Reviews. *SIGKDD*, 2004.

X. Hu, N. Sun, C. Zhang, and T.-S. Chua Exploiting Internal and External Semantics for the Clustering of Short Texts Using World Knowledge. *CIKM*, 2009.

S. Kim and E. Hovy. Determining the Sentiment of Opinions. *COLING*, 2004.

A. C. Konig and E. Brill. Reducing the Human Overhead in Text Categorization. *KDD*, 2006.

Z. Kozareva, E. Riloff, and E. Hovy. Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. *ACL*, 2008.

T. Li, Y. Zhang, and V. Sindhwani. A Non-negative Matrix Tri-factorization Approach to Sentiment Classification with Lexical Prior Knowledge. *ACL*, 2009.

B. Liu, M. Hu, and J. Cheng. Opinion Observer: Analyzing and Comparing Opinions on the Web. *WWW*, 2005.

B. Liu. Handbook Chapter: Sentiment Analysis and Subjectivity. Handbook of Natural Language Processing. *Marcel Dekker, Inc. New York, NY, USA*, 2009.

L.M. Manevitz and M. Yousef. One-class SVMs for Document Classification. *Machine Learning*, 2002.

Q. Mei, X. Ling, M. Wondra, H. Su, and C.X. Zhai. Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs. *WWW*, 2007.

X. Meng and H. Wang. Mining User Reviews: from Specification to Summarization. *ACL-IJCNLP*, 2009.

K. Murthy, T.A. Faruquie, L.V. Subramaniam, K.H. Prasad, and M. Mohania. Automatically Generating Term-frequency-induced Taxonomies. *ACL*, 2010.

H. Nishikawa, T. Hasegawa, Y. Matsuo, and G. Kikui. Optimizing Informativeness and Readability for Sentiment Summarization. *ACL*, 2010.

B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. *EMNLP*, 2002.

B. Pang and L. Lee. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with respect to Rating Scales. *ACL*, 2005.

B. Pang and L. Lee. *Opinion mining and sentiment analysis*. Foundations and Trends in Information Retrieval, 2008.

HH. Pang, J. Shen, and R. Krishnan Privacy-Preserving, Similarity-Based Text Retrieval. *ACM Transactions on Internet Technology*, 2010.

A.M. Popescu and O. Etzioni. Extracting Product Features and Opinions from Reviews. *HLT/EMNLP*, 2005.

H. Poon and P. Domingos. Unsupervised Ontology Induction from Text. *ACL*, 2010.

G. Qiu, B. Liu, J. Bu, and C. Chen. Expanding Domain Sentiment Lexicon through Double Propagation. *IJCAI*, 2009.

S. Roy and L.V. Subramaniam. Automatic Generation of Domain Models for Call Centers from Noisy Transcriptions. *ACL*, 2009.

B. Shi and K. Chang. Generating a Concept Hierarchy for Sentiment Analysis. *SMC*, 2008.

R. Snow and D. Jurafsky. Semantic Taxonomy Induction from Heterogenous Evidence. *ACL*, 2006.

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.

I. Titov and R. McDonald. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. *ACL*, 2008.

P. Turney. Thumbs up or thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *ACL*, 2002.

Y. Wu, Q. Zhang, X. Huang, and L. Wu. Phrase Dependency Parsing for Opinion Mining. *ACL*, 2009.

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

H. Yang and J. Callan A Metric-based Framework for Automatic Taxonomy Induction. *ACL*, 2009.

S. Ye and T.-S. Chua. Learning Object Models from Semi-structured Web Documents. *IEEE Transactions on Knowledge and Data Engineering*, 2006.

J. Yi, T. Nasukawa, W. Niblack, and R. Bunescu. Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. *ICDM*, 2003.

L. Zhuang,F. Jing, and X.Y. Zhu Movie Review Mining and Summarization *CIKM*, 2006.