Sentiment Aggregation using ConceptNet Ontology

Subhabrata Mukherjee and Sachindra Joshi
IBM India Research Lab
{subhabmu, jsachind}@in.ibm.com

Abstract

Sentiment analysis of reviews traditionally ignored the association between the features of the given product domain. The hierarchical relationship between the features of a product and their associated sentiment that influence the polarity of a review is not dealt with very well. In this work, we analyze the influence of the hierarchical relationship between the product attributes and their sentiments on the overall review polarity. ConceptNet is used to automatically create a product specific ontology that depicts the hierarchical relationship between the product attributes. The ontology tree is annotated with feature-specific polarities which are aggregated bottom-up, exploiting the ontological information, to find the overall review polarity. We propose a weakly supervised system that achieves a reasonable performance improvement over the baseline without requiring any tagged training data.

1 Introduction

In recent years there has been a huge surge of activity in the social networking sites, blogs and review sites. The voluminous amount of data generated is a goldmine of information for the retail brands to find out the customer needs, concerns and potential market segments. Sentiment analysis aims to mine this information to find out the popular sentiment about any product and its associated features.

Traditionally sentiment analysis has been posed as a text classification task on features derived from the given text. In the product review domain, the initial works in sentiment analysis focused on classifying the entire review as positive or negative using various word-based and phrase-based features (Turney *et al.*, 2003; Turney 2002; Kamps *et al.*, 2002; Hatzivassiloglou *et al.*, 2000; Hatzivassiloglou *et al.*, 2002). The more recent works focused on product feature extraction from a review and performing feature-specific sentiment analysis (Hu *et al.*, 2004; Mukherjee *et al.*, 2012). For example, the review, *The audio quality of my new phone is absolutely awesome but the picture taken by the camera is a bit grainy*, is positive with respect to the *audio quality* and negative with respect to the *camera*. However, once the feature-specific polarities are obtained, the works do not describe any systematic approach to aggregate the feature-specific polarities to obtain the overall review polarity. A naïve count-based feature-specific polarity aggregation will not work well for reviews having different features with diverse opinions. A bag-of-words based model will pick up *awesome* and *grainy* as the sentiment features and mark the overall review as *neutral*. One may argue that the *audio quality* is more important to a *cell phone* than the *camera* and hence the overall review polarity should be positive. While the feature-specific model associates sentiment to features, it cannot do a polarity aggregation in absence of feature association information to find the overall review polarity.

Let us consider the following review taken from Amazon.com which more clearly depicts the necessity of learning the hierarchical product-attribute relationship and associated sentiments.

I bought a Canon EOS 7D (DSLR). It's very small, sturdy, and constructed well. The handling is quite nice with a powder-coated metal frame. It powers on quickly and the menus are fairly easy to navigate. The video modes are nice, too. It works great with my 8GB Eye-Fi SD card. A new camera isn't worth it if it doesn't exceed the picture quality of my old 5Mpixel SD400 and this one doesn't. The auto white balance is poor. I'd need to properly balance every picture taken so far with the ELPH 300. With 12 Mpixels, you'd expect pretty good images, but the problem is that the ELPH 300 compression is turned up so high that the sensor's acuity gets lost (softened) in compression.

International Joint Conference on Natural Language Processing, pages 570–578,
Nagoya, Japan, 14-18 October 2013.

The above example depicts the complexity involved in analyzing product reviews. The review has a mix of good and bad comments about various features of the product. A flat classification model which considers all features to be equally important will fail to capture the proper polarity of the review. The reviewer seems happy with the *camera size, structure, easy use, video modes, SDHC support etc*. However, the *auto-white balance* and *high compression* leading to sensor acuity seem to disappoint him. Now, the primary function of a camera is to take good pictures and videos. Thus *picture, video quality, resolution, color balance etc*. are of primary importance whereas *size, video mode, easy use etc.*, are secondary in nature. The overall review polarity should be negative as the reviewer shows concerns about the most important features of the camera.

In this paper, we propose a weakly supervised approach to aggregate the sentiment about various features of a product to give the overall polarity of the review, without requiring expensive labeled training data. The approach is weakly supervised due to the requirement of ConceptNet (created by crowd-sourcing), a dependency parser and a sentiment lexicon.

The objectives of the paper can be summarized as:

1. Automatically learning the product-attribute hierarchy from a knowledge resource, where we leverage ConceptNet (Hugo *et al*., 2004) to learn the product *attributes, synonyms, essential components, functionalities etc*. and create a domain specific ontology tree

2. Discovering the various features of a product in the review and extracting feature-specific sentiment

3. Mapping the product features with their associated sentiments to the ontology tree and aggregating the feature-specific sentiments to determine the overall review polarity

2 Related Works

The initial works in sentiment analysis used bag-of-words features like unigrams, bigrams, adjectives *etc.* which gave way to the usage of phrase-based features like part-of-speech sequences (Ex: adjectives followed by nouns) (Turney *et al.*, 2003; Turney 2002; Kamps *et al.*, 2002; Hatzivassiloglou *et al*., 2000; Hatzivassiloglou *et al.*, 2002). These works did not consider the attributes or features of the underlying product domain in the review. A review may contain multiple features with a different opinion about each feature. This makes it difficult to come up with an overall polarity of the review. The latter works addressed this issue by focusing on feature-specific sentiment analysis.

Feature-specific sentiment analysis attempts to find the polarity of a review with respect to a given feature. Approaches like dependency parsing (Wu *et al.*, 2009; Chen *et al.*, 2010; Mukherjee *et al.*, 2012), joint sentiment topic model using LDA (Lin *et al.*, 2009) have been used to extract feature-specific expressions of opinion. Although these works extract the feature-specific polarities, they do not give any systematic approach to aggregate the polarities to obtain the overall review polarity.

Wei *et al.* (2010) propose a hierarchical learning method to label a product's attributes and their associated sentiments in product reviews using a Sentiment Ontology Tree (HL SOT). Although our work stems from a similar idea, it differs in a number of ways. The HL-SOT approach is completely supervised, requiring the reviews to be annotated with *product-attribute relations*, as well as *feature-specific opinion expressions*. The approach requires a lot of labeling information which needs to be provided for every domain. Also, the authors do not describe any elegant approach to aggregate the feature-specific polarities of the children nodes to obtain the overall review polarity.

In this work, we use ConceptNet (Hugo *et al*., 2004) as a knowledge resource to automatically construct a domain-specific ontology tree for product reviews, without requiring any labeled training data. ConceptNet relations have an inherent structure which helps in the construction of an ontology tree from the resource. ConceptNet has been used in information retrieval tasks in other domains (Guadarrama *et al.*, 2008; Kotov *et al.*, 2012). But there has been a very few works (Sureka *et al.*, 2010) in sentiment analysis using ConceptNet. Unlike the previous works, we present an approach to deal with noisy and one-to-many relations in ConceptNet as well as the myriad of relations and the ensuing topic drift. We also present a novel sentiment aggregation approach to combine the feature-specific polarities with ontological information to find the overall polarity of the review.

3 Ontology Creation from ConceptNet

Ontology can be viewed as a knowledge base, consisting of a structured list of concepts, relations and individuals (Estival *et al.*, 2004). The hierarchical relationship between the product attributes can be best captured by an *Ontology Tree*. Wei *et al.* (2010) use a tree-like ontology structure that represents the relationships between a product's *attributes* or *features*. They define a Sentiment Ontology Tree (SOT) where each of the non-leaf nodes of the SOT represents an attribute of a camera and all leaf nodes of the SOT represent sentiment (positive/negative) nodes respectively associated with their parent nodes.

We adopt a similar idea and consider an *Ontology Tree* for a product domain (say, *camera*) where the *feature nodes* (attributes like *body*, *lens*, *flash etc.*) are annotated with *feature-specific polarities* of the review.

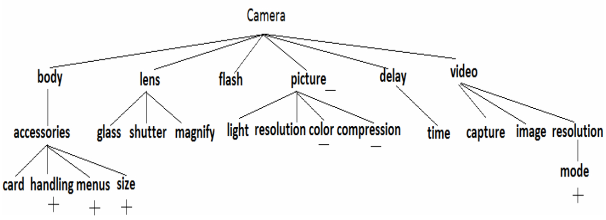

Figure 1. Snapshot of Camera Ontology Tree

The *feature nodes* in our *ontology tree* depict features of interest or attributes (Ex: *lens, flash, picture etc.*) of the given product (Ex: *camera*). The edges in the ontology tree depict the *relation type* connecting a feature with its parent. For example, a *lens* is a *partof* a *camera*, a *camera* is *usedfor* taking_*pictures*, *time_delay* is *derivedfrom time etc.* The *feature nodes* are annotated with *polarities* (+ and – denoting positive and negative sentiment, respectively) of the feature with respect to the review.

Figure 1 shows a snapshot of the ontology tree of a camera for the given example review in Section 1. The figure shows more positive feature-polarities than negative feature-polarities, but the review is still negative. This is because the feature polarities in the higher level of the ontology tree dominate those at a lower level, *i.e.* the importance of a feature dilutes with the increase in the ontology depth.

3.1 Domain Ontology Tree Creation

In this work, we leverage ConceptNet (Hugo *et al.*, 2004) to construct a domain-specific ontology tree for product reviews. ConceptNet is a very large semantic network of common sense knowledge which can be used to make various inferences from text. It is the largest, machine-usable common sense resource consisting of more than 250,000 propositions. Mining information from ConceptNet can be difficult as one-to-many relations, noisy data and redundancy undermine its performance for applications requiring higher accuracy (Smith *et al.*, 2004). However, we use ConceptNet for the following reasons:

1. The relational predicates in ConceptNet have an inherent structure suitable for building ontology. For example, relations like *partof, hasa, madeof* can be readily conceptualized as hierarchical relations.
2. ConceptNet has a closed class of well-defined relations. The relations can be suitably weighted and used for various purposes.
3. The continual expansion of the knowledge resource through crowd-sourcing incorporates new data and enriches the ontology.
4. Ontology creation using ConceptNet does not require any labeling of product reviews.

3.1.1 ConceptNet Relations

ConceptNet[1] has a closed class of 24 primary relations, expressing connections between various concepts.

camera UsedFor take_picture
camera IsA tool_for_take_picture
camera AtLocation store
tripod UsedFor keep_camera_steady
camera CapableOf record_image
camera IsA device
flash PartOf camera
lens AtLocation camera
tripod AtLocation camera_shop
camera IsA photo_device
cannon ConceptuallyRelatedTo camera
photograph ConceptuallyRelatedTo camera
picture ConceptuallyRelatedTo camera

Table 1. ConceptNet Relation Examples

We categorize the ConceptNet relations into 3 primary categories – *hierarchical* relations, *synonymous* relations and *functional* relations. *Hierarchical* relations represent parent-child relations and can be used to construct the tree top-down, as the relations are transitive. *Synonymous* relations help to identify related concepts. Thus similar nodes can be merged during tree construction. *Functional* relations help to identify the purpose or property of interest of the concept. The relation categorization helps to weigh various relations differently. Consider the case where the *functional* relation "a camera is *usedfor* taking_picture" may be of more interest to an individual than the hierarchical relation "a camera

[1]http://csc.media.mit.edu/docs/conceptnet/conceptnet4.html#relations

hasa tripod". Thus a product which takes good pictures but lacks a tripod will have a high positive polarity. This is, of course, subjective and can be used to personalize the ontology tree. The other advantage of relation categorization is to deal with one-to-many relations, as will be discussed in the next section.

Hierarchical	:	LocatedNear, HasA, PartOf, MadeOf, IsA, InheritsFrom
Synonymous	:	Synonym, ConceptuallyRelatedTo
Functional	:	UsedFor, CapableOf, HasProperty, DefinedAs

Table 2. ConceptNet Relation Type Categorization

3.1.2 Algorithm for Ontology Construction

Ontology construction from ConceptNet is hindered by the following obstacles:

1. One-to-many relations exist between the concepts. For example, the concepts *camera* and *picture* can be associated by relations like - camera *UsedFor* take_picture, camera *HasA* picture, picture *ConceptuallyRelatedTo* camera, picture *AtLocation* camera *etc.*
2. There is a high degree of *topic drift* during relation extraction. For example, the predicates camera *HasA* lens, lens *IsA* glass and glass *HasA* water places water at a high level in the ontology tree, although it is not at all related to camera.

The hierarchical relations in ConceptNet are much more definitive, have much less topic drift and can be used to ground the ontology tree. Hence, they are preferred over other relations during a relational conflict. In the above example, where *picture* is *ConceptuallyRelatedTo camera*, putting camera and picture at the same level will generate an incorrect ontology tree. The issue can be averted by preferring the hierarchical relation between camera and picture over the synonymous relation. The relational conflict is averted by ordering the predicate relations where hierarchical relations > synonymous relations > functional relations. In order to avoid topic drift, the ontology feature nodes extracted from ConceptNet are constrained to belong to a list of frequently found concepts in the domain, which is obtained from an unlabeled corpus.

In the first step of ontology construction, all the unlabeled reviews in the corpus are Part-of-Speech tagged and all *Nouns* are retrieved. The frequently occurring concepts are then added to the *feature set*. In the second step, the ConceptNet relations are partitioned into three disjoint sets *hierarchical*, *synonymous* and *functional*. The domain name is taken as the root of the *Ontology Tree*.

Input: Raw unlabeled corpus of product reviews and ConceptNet Knowledge Network

1. Part-of-speech tag the reviews and retrieve all *Nouns*. Let $\overline{N}$ be the set of all potential features.
2. A feature $n_i \in \overline{N}$ is considered relevant and added to the feature set N if $tf-idf(f_i) > \vartheta$, where ϑ is the corpus threshold
3. Let R be the set of all ConceptNet relations which is partitioned into the relation sets H (hierarchical), S (synonymous) and F (functional).
4. Every relation tuple $r_{ij}(f_i, f_j) \in R$ is assigned to one of the sets S, F or H with ties being broken as $H > S > F$
5. Construct the ontology tree $T(V,E)$ top-down. The root of the tree is taken as the domain name. Initially $V = \{domain\_name\}, E = \{\phi\}$.
6. Add a vertex v_j to V and an edge $e_{ij}(v_i, v_j)$ to E, $\forall r_{ij}(v_i, v_j) \in H$ *s.t.* $v_i \in V$ and $v_j \in N$
7. Merge v_j with v_i $\forall r_{ij}(v_i, v_j) \in S$ *s.t* $v_i \in V$ and $v_j \in N$
8. Add a vertex v_j to V and an edge $e_{ij}(v_i, v_j)$ to E, $\forall r_{ij}(v_i, v_j) \in F$ *s.t.* $v_i \in V$ and $v_j \in N$

Output : $T(V,E)$

Algorithm 1. Ontology Tree Construction from ConceptNet

The *hierarchical relation set* is taken first, and the tree is constructed recursively, such that the parent concept in any hierarchical relation is already in the tree and the child concept belongs to the set of frequently occurring concepts in the domain. The *synonymous relation set* is taken next, and similar concepts are merged recursively, such that one of the concepts in any synonymous relation is already in the tree and the other concept belongs to the frequently occurring *feature set*. In the last step, the *functional relation set* is taken and processed in the same way as the hierarchical relation set.

The constructed ontology tree depicts the product attributes in the domain and the different parent-child relations. The ontology creation does not require any labeled training data. Algorithm 1 shows the detailed steps for the ontology creation. Figure 1 shows a snapshot of the constructed ontology.

3.2 Feature Specific Sentiment Extraction

A review or a given sentence may contain multiple features with a different opinion regarding each feature. Given a sentence and a target feature, it is essential to obtain the polarity of the sentence with respect to the feature. For example the sentence, *"The movie had a nice plot but the acting was too shabby"*, is *positive* with respect to *plot* but *negative* with respect to *acting*.

In this work, we use the feature-specific sentiment extraction approach in Mukherjee *et al.* (2012), which do not need labeled review data for training. The authors use *Dependency Parsing* to capture the association between any specific feature and the expressions of opinion that come together to describe that feature.

Given a sentence *S*, let *W* be the set of all words in the sentence. Let *R* be the list of *significant* dependency parsing relations (like *nsubj, dobj, advmod, amod etc.*), which are learnt from a corpus. A Graph $G(W, E)$ is constructed such that any $w_i, w_j \in W$ are directly connected by $e_k \in E$, *if* $\exists R_l$ *s.t.* $R_l(r_i, r_j) \in R$. The *Nouns* are extracted by a POS-Tagger which form the initial feature set *F*. Let $f_i \in F$ be the target feature.

We initialize '*n*' clusters C_i, corresponding to each feature $f_i \in F$ *s.t.* f_i is the clusterhead of C_i. We assign each word $w_i \in S$ to the cluster whose clusterhead is closest to it. The distance is measured in terms of the number of edges in the shortest path, connecting any word and a clusterhead. Any two clusters are merged if the distance between their clusterheads is less than some threshold. Finally, the set of words in the cluster C_t, corresponding to the target feature f_t gives the opinion about f_t.

The words in the cluster C_t are classified with the help of a lexicon (*majority voting*) to find the polarity $p_t \in \{-1, 0, 1\}$ about the target feature f_t.

3.3 Sentiment Aggregation

Consider the camera review example in Section 1, and Figure 1 where the facets of the review are mapped to the camera ontology with their specific polarities. It can be observed that the product attributes at a higher level of the tree dominate those at the lower level. If a reviewer says something positive or negative about a particular feature, which is at a higher level in the ontology tree (say *picture*), it weighs more than the information of all its children nodes (say *light, resolution, color and compression*). This is because the parent feature abstracts the information of its children features. The feature importance is captured by the height of a feature node in the ontology tree. In case the parent feature polarity is neutral, its polarity is given by its children feature polarities. Thus the information at a particular node is given by its self information and the weighted information of all its children nodes. The information propagation is done bottom-up to determine the information content of the root node, which gives the polarity of the review.

Consider the ontology tree $T(V, E)$ where $V_i \in V$ is a *product attribute set*. The product attribute set V_i is represented by the tuple $V_i = \{f_i, p_i, h_i\}$, where f_i is a product feature, p_i is the review *polarity score* with respect to f_i and h_i is the height of the *product attribute* in the *ontology tree*. $e_{ij} \in E$ is an *attribute relation type* (*Section 3.1.1*) connecting $f_i \in V_i, f_j \in V_j$ and $V_i, V_j \in V$. Let V_{ij} be the j^{th} child of V_i .

The *positive sentiment weight* (PSW) and *negative sentiment weight* (NSW) of a vertex V_i are defined as,

$$PSW(V_i) = h_i \times p_i^+ + \sum_j PSW(V_{ij}) \times u_{ij}$$
$$NSW(V_i) = h_i \times p_i^- + \sum_j NSW(V_{ij}) \times u_{ij}$$

where $p_i^+ \in \{0,1\}$ and $p_i^- \in \{-1,0\}$.

The review polarity is given by the *expected sentiment-weight* (ESW) of the tree defined as,

$$ESW(root) = PSW(root) + NSW(root)$$

Consider Figure 1 and assume the edge-weights of the tree to be 1.

$$PSW(accessories) = 2 \times 0 + (1 \times 1 + 1 \times 1 + 1 \times 1) = 3$$
$$NSW(accessories) = 0, PSW(picture) = 0, PSW(video) = 1$$
$$NSW(picture) = 3 \times -1 + (-1 \times 2 - 1 \times 2) = -7$$
$$PSW(camera) = 4, NSW(camera) = -7, ESW(camera) = -3$$

Figure 2 shows a snapshot of the camera ontology tree annotated with positive and negative sentiment weights. Each feature node f_i is annotated with a tuple [p_i^+ , p_i^-] corresponding to its positive sentiment weight and negative sentiment weight respectively. Absence of a weight indicates that the feature node has a neutral sentiment. The figure depicts the importance of hierarchical learning as the negative sentiment weight of *picture*, at a higher level of the tree, dominates the positive sentiment weight of the other feature nodes at a lower level in the tree, resulting in the overall review polarity being negative.

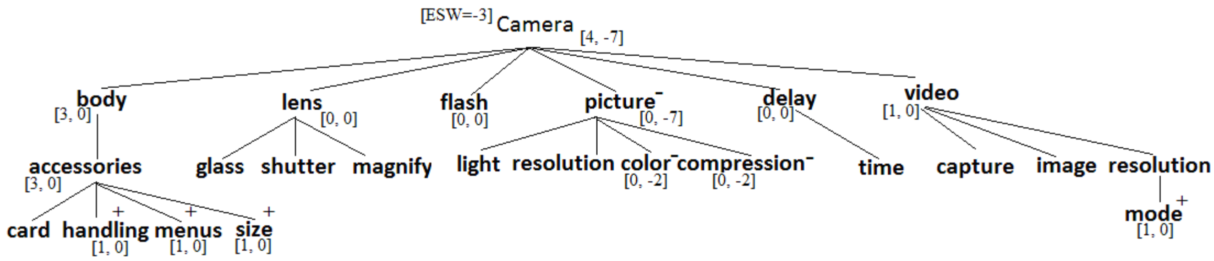

Figure 2. Snapshot of Camera Ontology Tree with Sentiment Weights

4 Experimental Evaluation

Analysis is performed in three domains corresponding to *automobile*, *camera* and *software*.

4.1 Dataset Preparation

Domain	Positive Reviews	Negative Reviews	Total Reviews
Automobile	584	152	736
Camera	986	210	1196
Software	1000	915	1915

Table 3. Dataset Statistics

The camera reviews are collected from Amazon.com and manually tagged as positive or negative. The automobile and software reviews[2] are taken from Blitzer *et al.* (2007). Table 3 shows the dataset statistics.

All the words are lemmatized in the reviews so that camera and cameras are reduced to the same root word camera.

Words like *hvnt, dnt, cnt, shant etc.* are replaced with their proper form in both our model and the baseline to capture negation.

4.2 Baselines

In this work, we consider three unsupervised baselines to compare the proposed approach.

1. Lexical Baseline: Lexical classification (Taboada *et al.*, 2011) is taken as the *first* baseline for our work. A sentiment lexicon is taken which contains a list of positive and negative terms. If the number of positive terms is greater than the number of negative terms, the review is considered to be positive and negative otherwise. The same approach is also used in our work while finding the polarity of the cluster representing the feature-specific opinion about a review. The lexical baseline considers all unigrams to be equally important, whereas we distinguish features by their position in the ontology hierarchy. This baseline model does not incorporate feature-specificity.

We experimented with three publicly available lexicons to obtain unigram polarities:

1. SentiWordNet 3.0 (Baccianella *et al.*, 2010)
2. General Inquirer (Stone *et al.*, 1966)
3. Bing Liu Lexicon (Hu *et al.*, 2004)

2. Corpus Feature-Specific Baseline: Tf-Idf measure is used to obtain the frequently occurring concepts in the domain from an unlabeled corpus. A feature-specific sentiment extraction model (Mukherjee *et al.*, 2012) is used to find the review polarity regarding each feature. A linear aggregation of the feature-specific polarities is done to obtain the overall review polarity. If the aggregation of the positive feature-specific polarities is greater than the aggregation of the negative feature-specific polarities, the review is considered to be positive and negative otherwise.

This model resembles the approach of LARA (Wang *et al.*, 2010) in a loose way, where the authors jointly learn the feature weights and feature-specific polarities.

3. ConceptNet and Corpus Feature-Specific Baseline: In this baseline, the features are extracted using ConceptNet and an unlabeled corpus using Algorithm 1. The feature set $\overline{F} = H \vee S \vee F$ is considered and the same feature-specific sentiment extraction model is used to aggregate all the feature-specific polarities in the set.

All the baselines lack sentiment aggregation (refer *Section 3.3*) using ontological information.

A simple *negation* handling approach is used both in our work and the baselines. A window of size 5 (Hu *et al.*, 2004) is taken and polarities of all the words appearing in the window starting from any of the negation operators *not, neither, nor* and *no* are reversed.

Table 4 shows the three baselines and the proposed approach with the different features used in the models.

[2] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

Models	Lexical	Corpus	ConceptNet	Sent. Aggr.
Lexical Baseline	Y			
Corpus Feature Specific Baseline	Y	Y		
Corpus and ConceptNet Feature Specific Baseline	Y	Y	Y	
Sent. Aggr. With Ontology Info.	Y	Y	Y	Y

Table 4. Models and Baselines

4.3 Results

Stanford Pos-Tagger[3] is used to part-of-speech tag the reviews to find the frequently occurring concepts (*Nouns*) in the domain. The ontology construction is done using ConceptNet 5[4]. The depth of the ontology tree is taken till level 4. The ontology depth has been empirically fixed. Further increase in depth leads to topic drift and domain concept dilution. Table 5 shows the number of frequently occurring concepts in the corpus, and the total number of nodes, leaf nodes and edges in the ontology tree for each domain.

Domains	Corpus Frequent Features	Ontology Nodes	Ontology Edges	Leaf Nodes
Automobile	268	203	202	76
Camera	768	334	333	148
Software	1020	764	763	208

Table 5. Ontology Tree Statistics

Table 6 shows the accuracy of the three lexical baselines in different domains in the dataset.

Lexicons	Automobile	Camera	Software
SentiWordNet 3.0	60.88	59.32	60.76
General Inquirer	65.70	68.15	66.14
Bing Liu Lexicon	64.43	63.65	69.38

Table 6. Lexical Baselines

Stanford Dependency Parser[5] is used to parse the reviews for dependency extraction during feature-specific sentiment analysis (refer *Section 3.2*). All the edge weights u_{ij} are taken to be 1. Table 7 shows the overall accuracy comparison of the proposed approach with the baselines. Bing Liu sentiment lexicon is used in all the approaches as it is found to deliver a better performance compared to the other lexicons in our model.

Models	Automobile	Camera	Software
Lexical Baseline (Bing Liu)	64.43	63.65	69.38
Corpus	68.34	65.25	72.54
ConceptNet + Corpus	70.19	67.15	74.74
ConceptNet + Corpus + Sent. Aggr.	**71.38**	**72.90**	**76.06**

Table 7. Overall Accuracy of All Models

Figure 3 shows the accuracy of different models on the positive and negative dataset in each domain.

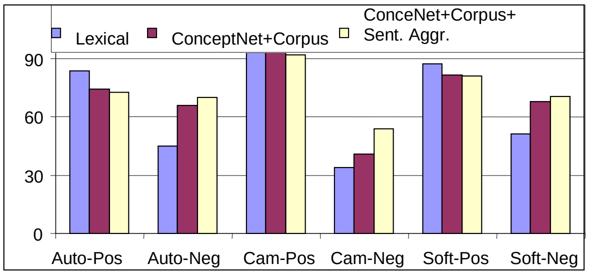

Figure 3. Positive and Negative Accuracy of Models in Each Domain

5 Discussions

In this section, we discuss the observations from the experimental results of using sentiment aggregation approach with ConceptNet Ontology.

1. Ontology Construction: The first part of our work outlines an approach to leverage ConceptNet to construct a domain-specific ontology for product reviews. It is a difficult task to evaluate the purity of any ontology. In our work, we only perform a qualitative analysis where the constructed ontology is found to contain most of the relevant concepts in the given domain with appropriate hierarchy.

It is observed that *75.75%* of the concepts in the automobile domain are mapped to some relevant concept in the corresponding product ontology; the corresponding figures for the camera and software domain being *43.49%* and

[3] http://nlp.stanford.edu/software/tagger.shtml
[4] http://conceptnet5.media.mit.edu/
[5] http://nlp.stanford.edu/software/lex-parser.shtml

74.90% respectively. In the camera domain, although the number of ontology feature nodes is much less than the frequently occurring concepts in the reviews, the proposed model performs much better than the baseline, which considers all features to be equally relevant. This shows that the ontology feature nodes capture concepts which are most relevant to the product and hence, makes a difference to the overall review polarity.

2. Lexical Baseline Performance: General Inquirer and Bing Liu sentiment lexicons outperform SentiWordNet in our dataset. Bing Liu sentiment lexicon was subsequently found to work better in our model than General Inquirer.

3. Corpus Feature-Specific Baseline: A significant accuracy improvement is observed over the lexical baseline due to the consideration of feature-specific polarities of relevant features mined from the frequently occurring concepts in the domain corpus.

3. ConceptNet and Corpus Feature-Specific Baseline: Incorporating ConceptNet information during the feature extraction process from the corpus improves the model performance. Only the features that frequently occur in the domain and form an important concept in the ontology hierarchy are retained.

4. Sentiment Aggregation: The model using sentiment aggregation approach by combining the feature-specific polarities with ontology information achieved the best accuracy in all the three domains.

5. Negative Opinion Detection: Reviews have much more explicit positive expressions of opinion than negative ones (Kennedy *et al.*, 2006; Voll *et al.*, 2007; Mukherjee *et al.*, 2012). This is because negative emotions are often very implicit and difficult to capture, as in sarcasm and thwarting. This is evident from Figure 3, where the lexical baseline attains a high accuracy on positive reviews in all the domains, but fares very poorly on negative reviews. The other two models, on the other hand, perform much better on the negative reviews. This shows that the ontology based sentiment extraction method is able to capture negative sentiment much more strongly. The model also paves the way for analyzing reviews which contain more positive expressions of opinion than negative ones, but are still tagged as negative; which cannot be captured by a feature-counting classifier.

6. Sentiment Ontology Tree Personalization: In this work, we have assumed all relations to be equally important, and thus considered the edge weights in the tree to be 1. However, the model allows the ontology tree to be personalized to suit the purpose of an individual and incorporate subjectivity in the reviews. If an individual prefers functional relations or use of certain features over its components, this information can be incorporated in the tree. This allows the general domain-specific ontology tree to be customized to an individual's interest.

6 Conclusions and Future Work

In this work, we outline an approach to combine the feature-specific polarities of a review with ontology information to give better sentiment classification accuracy. The proposed approach leverages ConceptNet to automatically construct a domain specific ontology tree. We performed experiments in multiple domains to show the performance improvement induced by the sentiment aggregation approach using ontology information over simple aggregation of feature-specific polarities.

The work is mostly unsupervised, requiring no labeled training reviews. The performance of the classifier is subject to the coverage of the lexicon and the accuracy of the feature-specific classifier.

The work also addresses the idea of personalizing a sentiment ontology tree to suit an individual's interest over specific features and parent-feature relations. This is also the first work, to the best of our knowledge, to discuss an approach to deal with reviews having majority positive (or negative) features but still tagged as negative (or positive). Reviews, of such kind, can be aptly handled using ontology information which captures the intrinsic specificities of product-feature relations in a given product domain.

References

A. Sureka, V. Goyal, D. Correa, and A. Mondal. 2010. Generating Domain-Specific Ontology from Common-Sense Semantic Network for Target-Specific Sentiment Analysis. In Fifth International Conference of the Global WordNet Association (GWC).

Alistair Kennedy and Diana Inkpen. 2006. Sentiment classification of movie and product reviews using contextual valence shifters. Computational Intelligence, 22(2):110–125, 2006.

Chen Mosha. 2010. Combining Dependency Parsing with Shallow Semantic Analysis for Chinese Opinion-Element Relation Identification. IEEE, pp.299-305.

Dustin Smith and Stan Thomas. 2004. Investigating ConceptNet. Ph.D Thesis. December 2004

Estival, D. Nowak, C. and Zschorn, A. 2004. Towards Ontology-Based Natural Language Processing. In Proceedings of 4th Workshop on NLP and XML.

Hu, Minqing and Liu, Bing. 2004. Mining and summarizing customer reviews, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Seattle, Washington, USA.

Hugo Liu and Push Singh. 2004. ConceptNet: A practical commonsense reasoning toolkit. BT Technology Journal, 22(4):211–226, October.

Jaap Kamps, Maarten Marx, Robert J. Mokken, and-Marten de Rijke. 2002. Words with attitude. In Proceedings of the 1st International Conference on Global WordNet, Mysore, India

John Blitzer, Mark Dredze, Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In Proceedings of Association of Computational Linguistics (ACL).

Kimberly Voll and Maite Taboada. 2007. Not all words are created equal: Extracting semantic orientation as a function of adjective relevance. In Proceedings of the 20th Australian Joint Conference on Artificial Intelligence.

Kotov, Alexander and Zhai, ChengXiang. 2012. Tapping into knowledge base for concept feedback: leveraging conceptnet to improve search results for difficult queries. In Proceedings of the fifth ACM international conference on Web search and data mining.

Lin, Chenghua and He, Yulan. 2009. Joint sentiment/topic model for sentiment analysis. In Proceedings of the 18th ACM conference on Information and knowledge management (CIKM).

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.

P.D. Turney and M.L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS), 21(4):315–346.

P.D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia

Sergio Guadarrama and Marta Garrido. 2008. Concept-Analyzer: A tool for analyzing fuzzy concepts. In Proceedings of IPMU.

Stefano Baccianella and Andrea Esuli and Fabrizio Sebastiani. 2010. SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining, LREC.

Stone, P.J., Dunphy, D.C., Smith, M.S., Ogilvie, D.M. and associates. 1966. The General Inquirer: A Computer Approach to Content Analysis. The MIT Press.

Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. Feature specific sentiment analysis for product reviews. Part 1, Lecture Notes in Computer Science, Springer 7181:475–487.

Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. WikiSent: Weakly Supervised Sentiment Analysis through Extractive Summarization with Wikipedia. In Procedings of Machine Learning and Knowledge Discovery in Databases - European Conference.

Taboada, Maite and Brooke, Julian and Tofiloski, Milan and Voll, Kimberly and Stede, Manfred. 2011. Lexicon-based methods for sentiment analysis. Computational Linguistics.

Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In Proceedings of EMNLP 2004, pp.412-418.

V. Hatzivassiloglou and J. Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity.

V. Hatzivassiloglou and K.R. McKeown. 2002. Predicting the semantic orientation of adjectives. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the ACL.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, Lide Wu. 2009. Phrase Dependency Parsing for Opinion Mining. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Wang, Hongning and Lu, Yue and Zhai, ChengXiang. 2010. Latent aspect rating analysis without aspect keyword supervision. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining

Wei, Wei and Gulla, Jon Atle. 2010. Sentiment learning on product reviews via sentiment ontology tree. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.