Graph Based Sentiment Aggregation using ConceptNet Ontology

Srikanth G Tamilselvam and **Seema Nagar** and **Abhijit Mishra** and **Kuntal Dey**
IBM Research, Bangalore and New Delhi, India
{srikanth.tamilselvam, senagar3, abhijimi, kuntadey}@in.ibm.com

Abstract

The *sentiment aggregation problem* accounts for analyzing the sentiment of a user towards various aspects/features of a product, and meaningfully assimilating the pragmatic significance of these features/aspects from an opinionated text. The current paper addresses the sentiment aggregation problem, by assigning weights to each aspect appearing in the user-generated content, that are proportionate to the strategic importance of the aspect in the pragmatic domain. The novelty of this paper is in computing the pragmatic significance (weight) of each aspect, using graph centrality measures (applied on domain specific ontology-graphs extracted from *ConceptNet*), and deeply ingraining these weights while aggregating the sentiments from opinionated text. We experiment over multiple real-life product review data. Our system consistently outperforms the state of the art - by as much as a F-score of 20.39% in one case.

1 Introduction

User-generated content accounts for a large fraction of the online content that is available today. The advent of web-platforms such as on online social networks (*e.g., Facebook*, *Twitter etc.*), blogs, discussion forums and product portals (*e.g.*, Amazon) has resulted in creation of a plethora of user-generated opinionated-content. The task of sentiment analysis focuses on analyzing such text, and deciphering user sentiment towards given products or their features, often referred to as *aspects*.

The task of sentiment aggregation builds upon sentiment analysis processes in general. While sentiment analysis (SA) aims to classify an opinion into positive or negative or neutral categories (in case of coarse-grained SA) or into more intricate categories (in case of fine-grained/dimensional SA), often enough, it considers the opinion in its entirety, and is agnostic of the aspect-specific sentiments expressed. However, in practical settings, opinions expressed by users (such as product reviews) often tend to focus on multiple aspects, not just one. The diversity of the aspects, does not allow the aspect-specific sentiment-polarity values to be just naively summed up for the purpose of obtaining an overall aggregated sentiment of a given user towards a given product.

As an example, let us consider the following text regarding a software application: *This app has a beautiful interface. It is not bug free though.* The first sentence here is a positive feel to it from the *interface* feature (aspect). The second sentence has a negative feel to it from the *accuracy (bug)* feature (aspect). Thus, a simple linear aggregation (sum of the individual polarities) of the features, will yield a overall neutral review polarity score. This is not necessarily accurate. While an interface is a necessary enabler for users to use a software, it is of imperative necessity that the software runs accurately, without posing problems of bugs. Perhaps, a reasonable interface with an error-free running platform, is more necessary for an application software, than to just have a beautiful interface and erroneous execution. So the overall polarity of this review should be negative.

Let us flip the above example to the following: *This app has a bad and unfriendly interface. The software is, however, stellar in terms of execution.* This time, the first sentence shows a negative intent, while the second shows positive. Again, clearly, the overall sentiment can be deemed positive, although a simple-minded sum of the individual aspects would yield a neutral value.

While in the previous examples the words such as *however* and *though* could potentially appear to act as discourse markers, such discourse markers will not appear in most of the cases. For instance, consider the following review example: *I hated the little hints they gave us hardcore NDers on the Dare to Play message board. I had extremely high hopes for this game–I adored Last Train to Blue Moon Canyon and waited AGES for this game. Now, besides the absurd level of difficulty in this game [it was seemingly VERY hard in my eyes–I could not make it the first time through without the assistance of a walkthrough], I LOVED it.* Clearly, while there are both positive and negative aspects in the review, the aggregate sentiment of the review is overall positive. And there is no discourse marker to simplify detecting the aggregated sentiment.

Assigning an overall review polarity requires a deeper aggregation of the sentiment polarity of each aspect. This involves not only understanding the sentiment purely from a natural language processing (NLP) standpoint, but also needs to account for the domain, and the pragmatic significance of the feature in the given domain.

Multiple works in the literature, such as (Hatzivassiloglou and Wiebe, 2000), (Turney, 2002), (Wu et al., 2009) and (Chen and Yao, 2010) have attempted to perform sentiment aggregation. One promising line of work explores usage of *domain ontologies* in order to factor for the *pragmatic value* of each aspect of the product. A recent work by (Mukherjee and Joshi, 2013) attempted to use *ConceptNet* (Liu and Singh, 2004), to learn the product attribute-hierarchy over *attributes*, *synonyms*, *essential components* and *functionalities*, and create a domain-specific ontology tree, using ConceptNet relationships across concepts. They subsequently map the sentiments associated with each feature of each given product to this ontology tree, and determine the overall aggregated sentiment as a weighted sum of these features, where the weights are computed as a function of proximity of the concept to the root node of the ontology.

Our work also uses domain ontologies for sentiment aggregation; however, we do not use the concept of ontology trees that has been used in the literature (Mukherjee and Joshi, 2013). We propose a novel approach, that extracts ontology-graphs from ConceptNet, around given themes, *e.g.*, *software*. The extracted concepts are assigned weights

using a measure of their *centrality* to the theme under consideration (*e.g.*, the centrality of *bug* given the theme *software*). Akin to Mukherjee and Joshi (2013), these weights are combined with the associated feature sentiments, and a weighted aggregation is carried out to obtain the final sentiment aggregation scores for each user review.

For experiments, we use the same datasets as used by (Mukherjee and Joshi, 2013). Our system is more effective compared to the rest of the literature: it outperforms the state of the art for all the domains, including a large F-score margin of 20.39% in one case. Amongst the four datasets we experiment with, *closeness centrality* often outperforms the other graph centrality measures we use, namely *betweenness centrality* and *PageRank*; however, *betweenness centrality* outperforms the rest in a few cases. These centrality measures are explained in Section 3.1.

Overall, we provide a novel graph-driven baseline over domain ontologies, for deeply ingraining pragmatic information of various aspects of product reviews. Experiments indicate that the performance of our system is consistent across datasets, and also it consistently outperforms the state of the art. The system is expected to provide insights to organizations in understanding overall user sentiments towards products, by analyzing user-generated natural language text content.

2 Related Work

Sentiment analysis (SA) has been an area of longstanding area of research. A seminal work was carried out by Hatzivassiloglou and McKeown (1997), attempting to identify the sentiment polarity orientation of adjectives, using conjunction constraints, using a four-step supervised learning algorithm. One school of research has conducted significant exploration towards SA from user generated content, and a large fraction of these works look at the social media such as Twitter. This includes works by Agarwal et al. (2011), Barbosa and Feng (2010) and Kouloumpis et al. (2011), and Opinion Finder at University of Pittsburgh (Wilson et al., 2005). Many recent works, such as Khan et al. (2015), Kolchyna et al. (2015), Le and Nguyen (2015), Severyn and Moschitti (2015), and Zimbra et al. (2016), have also investigated the sentiment analysis problem on user generated content. Recent systems are based on variants of deep neural network built on top of embed-

dings. A few representative works in this direction for sentiment analysis are based on Convolutional Neural Networks (CNNs) (dos Santos and Gatti, 2014; Kim, 2014; Tang et al., 2014), Recurrent Neural Networks (RNNs) (Dong et al., 2014; Liu et al., 2015) and combined architecture (Wang et al., 2016). A few works exist on using deep neural networks for sarcasm detection, such as by Ghosh and Veale (2016) that uses a combination of RNNs and CNNs.

Sentiment analysis for product reviews has been investigated since a long time, in the literature. An early work attempting to classify reviews into positive vs. negative was conducted by Tong (2001), generating sentiment timelines. This was followed by several works, that attempted to replace the bag-of-words based early models by more sophisticated feature driven models (such as lexical, syntactic and semantic features). Some noteworthy works in this space include Hatzivassiloglou and Wiebe (2000), Kamps et al. (2001), Turney (2002) and Turney and Littman (2003). However, neither did these works consider domain-specific information, nor did they account for users' views on the different aspects (features) of a given product - all of which are central to comprehend the overall aggregate sentiment of a user towards a product.

Subsequent works attempted to incorporate user sentiments towards specific product aspects (product features); few of these incorporated deeper NLP techniques, such as dependency parsing (Wu et al., 2009; Chen and Yao, 2010; Mukherjee and Bhattacharyya, 2012) and joint sentiment topic models using LDA (Lin and He, 2009). These works, however, do not provide methodical or robust approaches to combine the feature-specific sentiments, to form the aggregate review polarity of a given user towards a given product.

Mukherjee and Joshi (2013), which happens to be the work closest to ours, attempt to overcome this shortcoming, extending over Wei and Gulla (2010) and Sureka et al. (2010). Wei and Gulla (2010) propose a hierarchical learning method for labeling product attributes and the associated sentiments in product reviews, using a Sentiment Ontology Tree (HL SOT) with a supervised learning technique. However, the work requires feature-specific labeling, which is not practicable in real-life applications, as well as, falls short of proposing any elegant aggregation mechanism for integrating feature-specific sentiments. Sureka et al.

(2010) were among the first ones to use ConceptNet (Liu and Singh, 2004) in sentiment analysis. Mukherjee and Joshi (2013) borrow from the concept of sentiment ontology tree, extract feature-specific ontology using ConceptNet, and finally present an Expected Sentiment Weight based combination of the feature-specific polarities, relying upon (a) the feature-specific sentiment polarity and strength, and (b) the weight of the sentiment derived as a function of the distance of the concept phrase (feature) from the root of the ontology tree. Their system also accounts for noisy and one-to-many relations in ConceptNet, and topic drift.

We notice that, there is scope in research, to fully exploit the intricate inter-relationship of the non-independent concepts that are practically bound to arise in a real-life setting. For example, in a world of *camera*, the concepts of *lens* and *flash* cannot be completely independent. And yet, in the ontology-tree based approach, such interdependencies go uncaptured. Such examples are clearly seen in Figure 1 in Mukherjee and Joshi (2013). This motivates the need to better capture the intricate interdependencies of such concepts, in terms of interdependencies as well as the significance of such dependencies. We hence propose a graph-based approach, where graph vertex properties, including weights, derived from the connectivity structures, can accommodate for such factors. Our approach is weakly supervised, unlike most of the recent SA systems mentioned above, including the conceptually similar but supervised approach of Socher et al. (2013) who model RNN on sentiment treebank for sentiment aggregation. The system is easy to implement and deploy, and consistently outperforms the literature for all the sentiment aggregation benchmark datasets.

3 Why Graph based Solutions?

As discussed earlier, a sentiment aggregator should leverage an ontology structure to alleviate the lack of awareness of the inter-relationships between aspects. It is, however, essential for ontology-driven sentiment aggregators to be aware of all possible inter-relationships between the aspects appearing in the opinionated text. The problem with the existing method of *transforming the ontology into a tree structure and aggregation of sentiment in a bottom-up manner* (Mukherjee and Joshi, 2013) is that it assumes the relationships between aspects (which are essentially mapped

to concepts present in the ontology) to be hierarchical, thereby straight-away eliminating non-hierarchical relationships (like *metonymy*). As discussed earlier, concepts like "lens" and "flash" in camera domain do not share a parent-child relationship; it is thus impossible to find a connection between these two nodes in the ontology-tree. Moreover, as per Mukherjee and Joshi (2013), the intensity of sentiment expressed are aggregated from the leaf node towards the root. This does not allow sharing of sentiment related information between nodes at same levels (like "lens" and "flash"). Our proposed systems overcomes this problem by using the ontology-graph structure as it is, without performing any lossy transformation, unlike Mukherjee and Joshi (2013). We measure the pragmatic importance of the nodes of the ontology-graph through various centrality measures (discussed in Section 3.1), which helps our system decide how much sentiment-information can be shared across nodes during sentiment aggregation. We believe, a graph-based sentiment aggregation technique like ours offers a more natural way of sentiment aggregation that preserves all possible interrelationships amongst aspects.

3.1 Graph Centrality: Definitions

The centrality measures, the key constituents our approach, are explained below. The definitions are borrowed from the domain of graph theory.

Closeness Centrality: The closeness centrality (Bavelas, 1950) of a vertex in a connected graph indicates how central the vertex is to the overall graph structure. This is defined as the average length of the shortest paths between the given vertex and all other vertices in the given graph. If a given connected graph G comprises n vertices, then, the closeness centrality $C(v)$ of a vertex $v \in G$ is computed as

$$C(v) = \frac{n-1}{\sum\limits_{u} d(u,v)} \approx \frac{n}{\sum\limits_{u} d(u,v)}, \text{ if } (n \gg 1)$$

where $d(u,v)$ represents the shortest path distance between the vertices u and v.

Betweenness Centrality: For each pair of vertices in any connected graph, there exists at least one shortest path between the pair of vertices, such that, the number of edges constructing the path is minimized (in case of unweighted graphs), or, the total weight of the edges constructed is minimized (weighted graphs). The betweenness cen-

trality (Freeman, 1977) of a given vertex of a graph is defined as the number of such shortest paths passing through the vertex. For a given vertex v, this is computed as

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from vertex s to vertex t and $\sigma_{st}(v)$ of these pass through vertex v.

PageRank: The PageRank (Brin and Page, 1998) of a given vertex in a graph denotes the stationary probability of a random walk with restarts to arrive on the given vertex. For a given vertex v_i on a given graph G constituting of n vertices, the PageRank of the vertex $P(v_i)$ is computed as

$$P(v_i) = \alpha \sum_j u_{ji} \frac{v_j}{L_j} + \frac{1-\alpha}{n}$$

where $L_j = \sum\limits_i u_{ji}$ is the number of neighbors of vertex j, and α, the damping factor, represents the probability of the random walk to continue.

4 Our Approach: Sentiment Aggregation using Graph Centrality

Our approach consists of two steps for assigning an overall polarity to a review for a product, based on the polarities expressed for individual aspects.

- The first step computes the *pragmatic significance* score for each aspect of a product, based on the graph centrality metrics obtained from domain specific ontology-graphs constructed from ConceptNet.

- The second step aggregates the polarities of different aspects to get one overall polarity for a review text, based on the scores from the earlier step and the polarities expressed for each aspect.

We provide the details of our approach below.

4.1 Ontology Graph Construction

We first construct the *ontology graph*, for each given concept/domain. Domain ontology captures intricate relationships and dependencies among different aspects of a product. We exploit the domain ontology in constructing a rich graph where nodes represent aspects and edges connecting them represent relationships among them. The

graph representation rightly captures all the dependencies among the aspects as well as the complicated relationships among them.

The graph is constructed as follows. The concept/domain is identified by a *seed word* such as, *camera, automobile, etc.*. A vertex is constructed for the seed word, and is marked as *unexplored* in the set of graph vertices. This is added to $\mathcal{V}$, the set of vertices in the ontology graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For each *unexplored* vertex v in the graph, each concept in ConceptNet that has at least one relationship with v, a vertex u is added to the vertex set $\mathcal{V}$ and is marked as *unexplored*, as well as, an edge (u, v) is added to $\mathcal{E}$. At this stage, the vertex v is marked as *explored*. The ontology graph creation algorithm completes, when all vertices in $\mathcal{V}$ are marked as *explored*. We provide a maximum graph distance cut-off of n (a given number), where n is number of edges on the minimum-length path to reach from the concept to the seed word in the graph. Algorithm 1 provides the details of the ontology graph creation algorithm.

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

Table 1: Categorization of ConceptNet Relationship Types

Three types of ConceptNet relationships are used to form the edges, shown in Table 1. *Hierarchical* relationships represent parent-child relationships of concepts. *Synonymous* relationships are used to identify related concepts. *Functional* relationships are used to identify the purpose or property of interest of the given concept.

4.2 Graph Centrality Computation

This step computes the centrality of each concept appearing in the domain ontology graph constructed in the earlier step. For computing pragmatic significance score for each aspect, we employ graph centrality measures. Specifically, we propose to use centrality metrics such as closeness centrality, betweenness centrality and page rank, since these centrality measures captures significance of a node from different perspectives. In

---

Algorithm 1 ONTOLOGY GRAPH CREATION

1: *function* **OntologyGraphCreate():**
2: $\mathcal{V} \leftarrow$ domain seed word as vertex s
3: **REM** *E.g.*: kitchen, automobile, software, camera *etc.*
4: mark s as *unexplored*
5: $\mathcal{E} \leftarrow \phi$
6: **while** there exists at least one *unexplored* vertex $v \in \mathcal{V}$ **do**
7:     **for** $u \in neighbors(v)$ **do**
8:         (**REM** neighbors(e) includes all vertices that have Hierarchical, Synonymous and Functional relationships with vertex e)
9:         **if** graph_distance(s,u) $\leq n$ **then**
10:             $\mathcal{V} \leftarrow \mathcal{V} \cup u$
11:             mark u as *unexplored* if not already explored previously
12:             $\mathcal{E} \leftarrow \mathcal{E} \cup (u, v)$
13:         **end if**
14:         mark v as *explored*
15:     **end for**
16: **end while**
17: **Output:** $\mathcal{G} \leftarrow (V, E)$

---

computation of the different centrality metrics we do not consider the type of relationship on an edge. An example of a closeness centrality graph, has been provided in Figure 1, for the domain *camera*.

4.3 Feature/Aspect-Specific Sentiment Computation

A user's opinion (review) could constitute of multiple aspects (features) of a given product, and different sentiment (opinion) polarities with respect to each aspect. The process of overall user sentiment analysis, mandates understanding the user's sentiment towards each of these aspects. To determine the sentiment polarity expressed by a user towards each aspect (feature), we perform *dependency parsing* of each review, to associate a given aspect of the review, with the opinion of the user towards the given feature, expressed in the text.

Let R be a user review towards a product. Let W be the words constituting the review R. Following the approach of (Mukherjee and Bhattacharyya, 2012), we perform dependency parsing, and obtain D, the set of significant dependency relations in the corpus (*e.g.*, *nsubj, amod, dobj, etc*). For each dependency D_l where $D_l(d_i, d_j) \in D$, a graph $G(W, E)$ is constructed
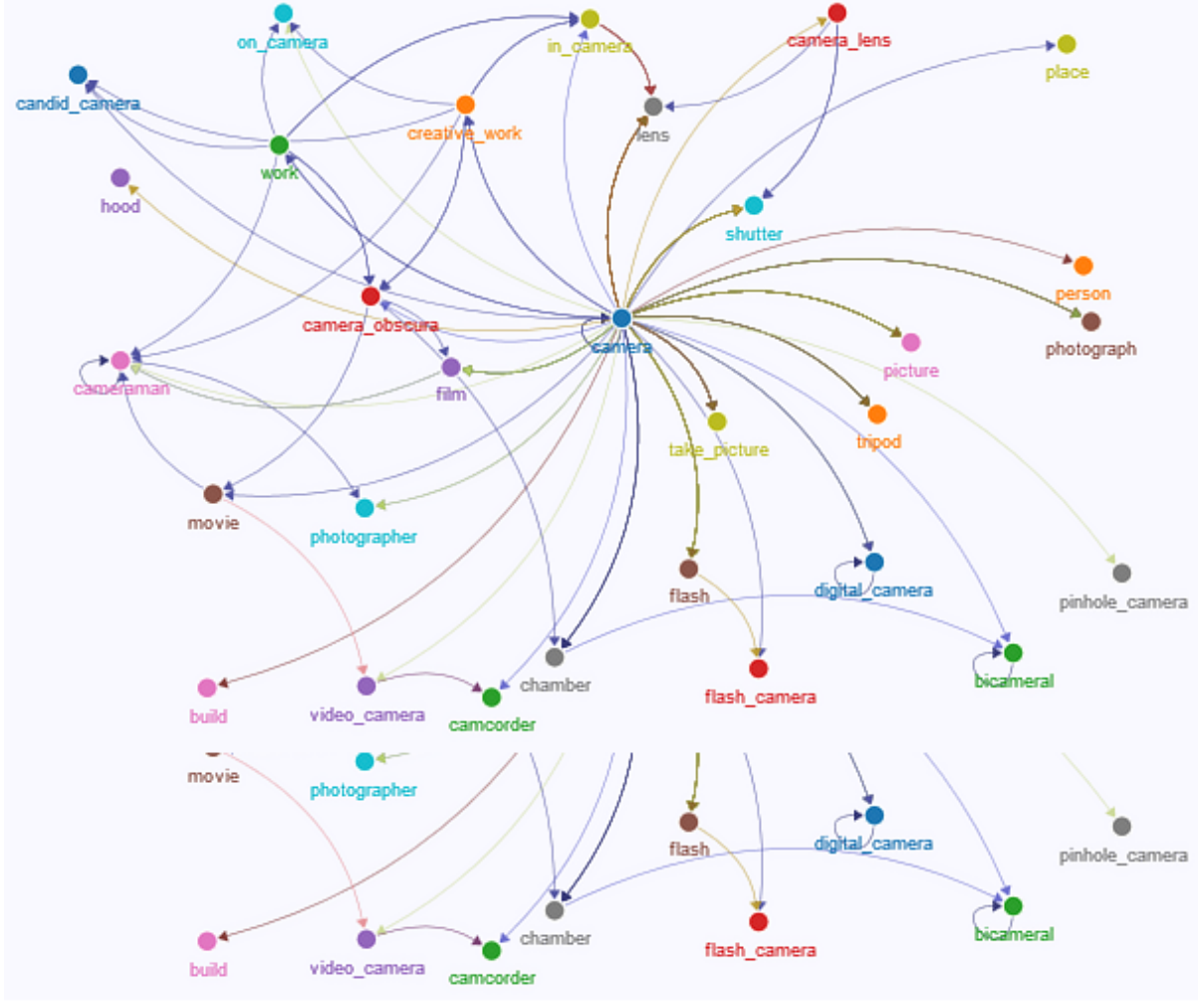
Figure 1: Depiction of closeness centrality measure for Camera Ontology

s.t. all $(w_i, w_j) \in W$ are connected by edge $e_k \in E$.

A PoS tagger is used to extract the entities (nouns). These entities are used as the initial feature (aspect) set $f_i \in F$. For each feature f_i, a cluster C_i is initiated, where f_i acts as the cluster head of C_i. Each word $w \in W$ occurring in review R, is assigned to the cluster having closest cluster head. The "closest" distance is measured, using the number of edges in the shortest path, that connects the word to the closest cluster head. Two clusters are merged, if the distance between the two cluster heads are less than a given threshold.

The set of words W_i belonging to each cluster C_i, are used to determine the user's opinion about feature f_i. This is attained by conducting a simple *majority voting* of the sentiment values of the individual words $w_i \in W_i$, using sentiment lexicons. A final aspect-specific sentiment score is produced, as -1 for negative, 0 for neutral and $+1$

for positive.

Also note that, we use the simple negation handling framework that was also adopted by (Mukherjee and Joshi, 2013). We reverse the sentiment polarities of all the words appearing within a window of size 5 (Hu and Liu, 2004), starting from any of the negation operators *not*, *nor*, *neither* and *no*.

4.4 Sentiment Polarity Aggregation

In this step, we aggregate polarities across all the aspects, to assign overall polarity to a review text. We define the overall polarity of a text as a weighted sum of (a) the sentiment polarity expressed by the user towards each aspect (feature) of the product the review text, and (b) the pragmatic significance of the aspect in the given domain, reflected in the graph centrality measures. These two factors ensure that the final polarity value for a given review text aggregates polar-

ity values across different aspects, with adequate weightage that the aspect requires.

Let R be a review text, for a product P which has M aspects. Let m_i and $m_i{}^p$ represent i^{th} aspect, and it's sentiment polarity as computed as described earlier. Let c_{m_i} represent the centrality score for an aspect m_i computed using the ontology graph for the domain which P belongs to. The polarity value $p(sum)$ for a review text R is found by computing the polarity value, as the following:

$$p(sum) = \sum_{i=1}^{M} m_i{}^p \times c_{m_i}$$

Finally, the aggregated sentiment polarity S is assigned as:

$$\begin{cases} S = Positve & if\, p(sum) > 0 \\ S = Negative & if\, p(sum) < 0 \\ S = Neutral & if\, p(sum) = 0 \end{cases}$$

To find the aggregate sentiment of all users for a given product, we again opt for a *majority-voting* strategy. The overall methodology is presented in Algorithm 2.

5 Datasets and Ontologies

Dataset from four different domains corresponding to *automobile*, *camera*, *kitchen* and *software* are used for experiments. The camera reviews are collected from Mukherjee and Joshi (2013). The automobile, kitchen and software reviews are taken from Blitzer et al. (2007). Table 2 shows the dataset statistics.

Domain	Positive Reviews	Negative Reviews	Total Reviews
Automobile	584	152	736
Camera	986	210	1196
Kitchen	1001	1000	2001
Software	1000	915	1915

Table 2: Dataset Statistics

Note that, akin to Mukherjee and Joshi (2013), all the words have been lemmatized in the reviews, which ensures that all the terms such as *camera* and *cameras* are treated as the root word *camera*. Further, words such as *hvnt*, *dnt* have been replaced to their original forms.

In the ontology graph construction process, we keep adding *unexplored* vertices to the vertex set,

Domain	Corpus Frequent Features	Ontology Nodes	Ontology Edges
Automobile	132	114	778
Camera	986	979	1280
Kitchen	767	670	10629
Software	150	135	842

Table 3: Ontology-graph Statistics

as long as, there is at least one edge between the corresponding concept to an existing vertex in the vertex-set, of one of the types *functional*, *hierarchical* or *synonymous*. However, we restrict to adding vertices such that the maximum distance between the *seed word* and the newly added concept remains less than a given threshold n. We empirically fix $n = 4$, which practically provides a sufficiently large number of concepts that are realistically related to the concept of the seed word. Higher values of n lead to domain concept delusion and topic drift. Table 3 presents statistics of the ontology graphs extracted for four domains.

6 Experiments

6.1 Tools and Resources

We use several well-known tools and resources.

- For PoS tagging, we use Stanford NLP Toolkit[1]. PoS tagging is carried out to tag the user reviews, which in turn is used to identify the entities (noun concepts) in the reviews.

- For ontology construction for the domains, we use ConceptNet 5[2].

- To compute centrality measures of the ontology graphs, we use the graph tool **R**[3].

- For dependency parsing of the user reviews, we use Stanford Dependency Parser[4].

- For sentiment lexicons, we experiment with SentiWordNet (Baccianella et al., 2010) and Bing Liu sentiment dictionary (Hu and Liu, 2004). Although we report our results only for the Bing Liu sentiment dictionary for the

---

[1]http://nlp.stanford.edu/software/tagger.shtml
[2]http://conceptnet5.media.mit.edu
[3]https://www.r-project.org
[4]http://nlp.stanford.edu/software/lex-parser.shtml

Algorithm 2 THE OVERALL APPROACH

1: $\mathcal{G}(\mathcal{V}, \mathcal{E}) \leftarrow$ OntologyGraphCreate()

2: **for** each vertex $v \in \mathcal{G}$ **do**
3:     centrality$(v) \leftarrow$ centrality measure value of vertex v in graph $\mathcal{G}$
4: **end for**

5: $total\_senti \leftarrow 0$
6: **for** each user review $\mathcal{R}$ **do**
7:     extract entities $\mathcal{T}$ from $\mathcal{R}$
8:     perform dependency parsing of $\mathcal{R}$
9:     $user\_senti \leftarrow 0$
10:     **for** each dependency $\mathcal{D}$ where $f = headword(\mathcal{D}) \in$ the full feature/aspect set $\mathcal{F}$ **do**
11:         (**REM** Each headword represents an aspect of the review)
12:         $dep\_senti(f) \leftarrow 0$
13:         **for** each word-concept $w \in \mathcal{D}$ **do**
14:             $dep\_senti(f) \leftarrow dep\_senti(f) + Sentiment(w)$
15:         **end for**
16:         **if** $f \in \mathcal{V}$ **then**
17:             $user\_senti \leftarrow user\_senti + dep\_senti(f) \times centrality(f)$
18:         **end if**
19:     **end for**
20:     $user\_senti\_set \leftarrow user\_senti\_set \cup < \mathcal{R}, user\_senti >$
21:     $total\_senti \leftarrow total\_senti + user\_senti$
22: **end for**
23: **Output:** $total\_senti$, $user\_senti\_set$

24: *function* **Sentiment(Word Concept w)**:
25: return Sentiment_Dictionary_Lookup(w) (**REM** SentiWordNet, Bing Liu *etc.*)

---

sake of brevity, we observe similar performances using SentiWordNet also.

6.2 Results

We establish the first baseline of our work, using the *lexical classification* based approach of (Taboada et al., 2011). In this approach, a sentiment lexicon is used as a reference, that consists of words having positive and negative sentiment polarities. In a given review, if the total number of positive terms is higher compared to the total number of negative terms, the review is considered positive, and is considered negative if the opposite holds true. The baseline does not incorporate the feature (aspect) specific approach. We modify the approach to associate the lexical terms with the aspects (features), and thus obtain a feature-specific lexical sentiment. We subsequently aggregate these sentiments, to obtain improved baseline results. For sentiment lexicon, we empirically explore with SentiWordNet and Bing Liu sentiment dictionary.

We further compare our work against the reported approach for the same task by (Mukherjee and Joshi, 2013) which also uses ConceptNet, and has an approach similar to ours. However, as mentioned earlier, they consider ontology as a *tree* while we construct a *graph*. Also, they assign pragmatic weights to each aspect present in the review, using the height (distance) of the aspect from the root (*seed word*) of the ontology tree they construct, while we use graph centrality measures.

Table 4 illustrates the results we obtain with different approaches. The performance of the systems are reported in terms of accuracy (to ensure direct comparison with previous work) and **weighted F1-score** (to tackle class-imbalances). We report the results of the lexical baseline by (Taboada et al., 2011) using Bing Liu sentiment dictionary baseline, the results of Mukherjee and

Models	Automobile		Camera		Kitchen		Software	
	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.
Lexical Baseline (Bing Liu)	73.45	64.43	79.74	63.65	66.77	67.11	66.04	69.38
Hierarchical Aggr. (Mukherjee et al., 2013)	71.48	70.23	81.22	70.38	67.28	67.62	70.19	70.28
Aggr. *Closeness* (Our approach)	75.93	**73.85**	**84.68**	**74.00**	85.66	**72.96**	**70.87**	**70.52**
Aggr. *Betweenness* (Our approach)	**76.47**	72.91	82.68	73.16	**87.67**	71.61	68.79	69.10
Aggr. *PageRank* (Our approach)	75.96	73.68	82.99	71.56	85.2	71.31	68.87	69.60
McNemar Significance Test (p)	< 0.0001		< 0.0001		< 0.0001		< 0.0001	

Table 4: Overall F1 score and accuracy (in %) of all models across all domains. For all domains, the performance improvements obtained using *Closeness* centrality measure over that reported in the literature (Mukherjee et al., 2013) are statistically significant (with $p << 0.05$), as confirmed by McNemar test.

Joshi (2013), and our results, using the three different graph centrality measures. As observed, all our proposed centrality based approaches outperform the baseline. The *closeness* centrality measure performs the best, with statistically significant improvement ($p << 0.05$) observed over the system reported by Mukherjee and Joshi (2013). Other graph based approaches also show improvement, except for the *software* domain.

7   Conclusion and Future Work

In this paper, we performed sentiment aggregation as a combination of user sentiments, analyzed towards multiple aspects/features of a product, from user-generated content. The novelty of this work was in deeply ingraining the sentiment weight of each entity derived from the user generated content, and pragmatic significance of the entity in the domains that was obtained by using a graph-structured ontology. We observe a consistently high performance of our system across all the keywords that we experiment with. We outperform the state of the art by a F-score of 3.02%, 3.46%, 20.39% and 0.68%, for *automobile*, *camera*, *kitchen* and *software* respectively. Further, the effectiveness of our system often increases by using *closeness centrality* over the other graph centrality measures such as *betweenness centrality* and *PageRank*, although *betweenness centrality* does outperform the rest of the methods in some cases. In future, we would like to improve the current technique to include the intensity of sentiment bearing words appearing in the reviews. Integrating lexico-semantic knowledge acquired through concept-embeddings learned from ontology structures in the aggregation step is also a future work. Our system will have significant real-life impact in helping organizations understand overall user sentiment towards products, on e-commerce sites

(*e.g.*, Amazon) as well as online social networks, discussion forums and blogs.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*. Association for Computational Linguistics, pages 30–38.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*. volume 10, pages 2200–2204.

Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, pages 36–44.

Alex Bavelas. 1950. Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America* 22(6):725–730.

John Blitzer, Mark Dredze, Fernando Pereira, et al. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*. volume 7, pages 440–447.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30(1):107–117.

Mosha Chen and Tianfang Yao. 2010. Combining dependency parsing with shallow semantic analysis for

chinese opinion-element relation identification. In *Universal Communication Symposium (IUCS), 2010 4th International*. IEEE, pages 299–305.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *ACL (2)*. pages 49–54.

Cícero Nogueira dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING*.

Linton C Freeman. 1977. A set of measures of centrality based on betweenness. *Sociometry* pages 35–41.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of NAACL-HLT*. pages 161–169.

Vasileios Hatzivassiloglou and Kathleen R McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 174–181.

Vasileios Hatzivassiloglou and Janyce M Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 299–305.

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*. ACM, pages 168–177.

Jaap Kamps, Maarten Marx, Robert J Mokken, and Marten de Rijke. 2001. *Words with attitude*. Citeseer.

Aamera ZH Khan, Mohammad Atique, and VM Thakare. 2015. Combining lexicon-based and learning-based methods for twitter sentiment analysis. *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)* page 89.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1746–1751.

Olga Kolchyna, Thársis TP Souza, Philip Treleaven, and Tomaso Aste. 2015. Twitter sentiment analysis. *arXiv preprint arXiv:1507.00955* .

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! *ICWSM* 11:538–541.

Bac Le and Huy Nguyen. 2015. Twitter sentiment analysis using machine learning techniques. In *Advanced Computational Methods for Knowledge Engineering*, Springer, pages 279–289.

Chenghua Lin and Yulan He. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, pages 375–384.

Hugo Liu and Push Singh. 2004. Conceptneta practical commonsense reasoning tool-kit. *BT technology journal* 22(4):211–226.

Pengfei Liu, Shafiq R Joty, and Helen M Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*. pages 1433–1443.

Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. Feature specific sentiment analysis for product reviews. *Computational Linguistics and Intelligent Text Processing* pages 475–487.

Subhabrata Mukherjee and Sachindra Joshi. 2013. Sentiment aggregation using conceptnet ontology. In *IJCNLP*. pages 570–578.

Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, pages 959–962.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, Christopher Potts, et al. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. volume 1631, page 1642.

Ashish Sureka, Vikram Goyal, Denzil Correa, and Anirban Mondal. 2010. Generating domain-specific ontology from common-sense semantic network for target specific sentiment analysis. In *Proceedings of the fifth international conference of the Global WordNet Association. Mumbai, India*. pages 1–8.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics* 37(2):267–307.

Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. 2014. Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. pages 208–212.

Richard M Tong. 2001. An operational system for detecting and tracking opinions in on-line discussion. In *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification*. volume 1, page 6.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 417–424.

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

Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2016. Dimensional sentiment analysis using a regional cnn-lstm model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 225–230.

Wei Wei and Jon Atle Gulla. 2010. Sentiment learning on product reviews via sentiment ontology tree. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 404–413.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Opinionfinder: A system for subjectivity analysis. In *Proceedings of hlt/emnlp on interactive demonstrations*. Association for Computational Linguistics, pages 34–35.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational Linguistics, pages 1533–1541.

David Zimbra, M Ghiassi, and Sean Lee. 2016. Brand-related twitter sentiment analysis using feature engineering and the dynamic architecture for artificial neural networks. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, pages 1930–1938.