Expressing Opinion Diversity

Andreea Bizău
Faculty of Mathematics and Computer Science
Babeș-Bolyai University
Cluj-Napoca, Romania
andreea.bizau@gmail.com

Delia Rusu
Artificial Intelligence Laboratory
Jožef Stefan Institute
Ljubljana, Slovenia
delia.rusu@ijs.si

Dunja Mladenić
Artificial Intelligence Laboratory
Jožef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

ABSTRACT

The focus of this paper is describing a natural language processing methodology for identifying opinion diversity expressed within text. We achieve this by building a domain-driven opinion vocabulary, in order to be able to identify domain specific words and expressions. As a use case scenario, we consider Twitter comments related to movies, and try to capture opinion diversity by employing an opinion vocabulary, which we generate based on a corpus of IMDb movie reviews.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: *Text analysis*.

General Terms

Algorithms, Design.

Keywords

Opinion mining, natural language processing, social networks.

1. INTRODUCTION

Information is expressed on the Web under a variety of forms, some of them more formal and standardized, like news articles, others more spontaneous, ad-hoc, like blogs or microblogs. One challenge is to tap into these sources, and allow for a diverse representation of information on the same topic, presenting different points of view, opinions, arguments.

In this work we are describing a natural language processing methodology for discovering the diversity of opinions expressed within text, which we deem to be an essential step to expressing and presenting diverse information on the Web. In this context, we consider an *opinion* as a subjective expression of sentiments, appraisals or feelings, and *opinion words* as a set of keywords/phrases used in expressing an opinion. As such, the *orientation* of an opinion word indicates whether the opinion expressed is positive, negative or neutral, while the totality of opinion words forms an *opinion vocabulary*. While opinion words can be analyzed in their base form (describe and convey the opinion directly) and comparative form (convey the opinion indirectly, by comparison with other entities), this research focuses only on base type opinion words.

In the context of the ever expanding world of social media and user generated content, instant access, world-wide coverage and diversity of perspective are the norm of the information flow. As an application of our approach, we propose to study the movie domain. There is a strong user interest in watching, tracking and discussing movies, generating highly diverse opinion content. Movies are subject to a variety of classifications, expanding the field of analysis. Moreover, the lifespan of a movie topic is longer than for usual topics, thus introducing a temporal dimension that can be further explored. Nowadays, accessing and assessing the public opinion has taken on a new form. Social networking encourages the exchange of information and sharing of opinions between individuals, friends and communities. Therefore, in our case study we directly address movie comments, as posted on Twitter, a popular social networking and microblogging website, and aim at identifying the diversity of opinions expressed in tweets related to movies. We determine a variety of polarized opinion words about a certain movie, and use these word frequency counts to obtain an overall aggregated opinion about the movie. Moreover, we can observe variations in opinions over time, related to a certain movie, by comparing the word frequency counts obtained from tweets belonging to a time interval (e.g. an hour, day, week).

The paper is structured as follows: in Section 2 we describe our algorithm for constructing a domain-driven opinion vocabulary, while Section 3 presents the Twitter movie comments use-case. The last section of the paper is dedicated to conclusions and future work.

2. DOMAIN DRIVEN OPINION VOCABULARY

We start from the idea that expressing opinions is dependent on the topic's context and we focus on the role of adjectives as opinion indicators; in the future we plan to broaden this line of work by including verbs and adverbs. The starting point is represented by a domain-specific corpus, from which we determine a small number of seed opinion words that we further extend, thus forming a domain-driven opinion vocabulary.

There are three main approaches to constructing an opinion vocabulary: manual, dictionary based and corpus based. The manual approach is not really in line with our work, as we are considering automatic, scalable approaches. The dictionary based approach provides a simple and efficient way of obtaining a good vocabulary. SentiWordNet [3] is a publicly available lexical resource. It provides tags of all WordNet [4] synsets with three numerical scores (objective, positive, negative), offering a general opinion vocabulary with good coverage. However, the dictionary-based approach cannot account for the domain specific orientation of words, nor can it identify domain specific words and expressions. As an example, consider the word *unpredictable*. In most situations it will express an undesirable quality (e.g. unpredictable car behavior), thus its orientation will be negative; but in the movie domain, an unpredictable plot is something desired and indicates a positive opinion. In order to account for domain specificity, we decided to employ a corpus based approach.

V. Hatzivassiloglou et al [6] showed the relevance of using connectives in gathering information about the orientation of conjoined adjectives. They emphasized that conjoined adjectives are of the same orientation, for most connectives, *but* reversing the relationship. The connectives are conjunctions used to join one or more adjectives together. In our algorithm we used a subset of the possible conjunctions (*and*, *or*, *nor*, *but*, *yet*), that cover many common syntactic patterns and are easier to correlate with the adjectives that they connect.

Other lines of research, like S.-M. Kim and E. Hovy [7] try to identify opinion expressions together with their opinion holder starting from a word seed list and use the WordNet synsets to determine the strength of the opinion orientation for the identified opinion words. M Gamon and A. Aue [5] extend the Turney-style [9] approach of assigning opinion orientation to the determined candidate words, working under the assumptions that in the opinion domain, opinion terms with similar orientation tend to co-occur, while terms with opposite orientation do not tend to co-occur at sentence level.

V Jijkoun et al [10] propose a different style of approach, by starting from an existing lexicon (clues) and focusing it. They perform a dependency parsing on a set of relevant documents, resulting in triplets (clue word, syntactic context, target of sentiment) that represent the domain specific lexicon. H. Kanayama and T. Nasukawa [11] apply the idea of context coherency (same polarity tend to appear successively) to the Japanese language. Starting from a list of polar atoms (minimum syntactic structure specifying polarity in a predicative expression), they determine a list of domain specific words using the overall density and precision of coherency in the corpus. Sinno Jialin Pan et al [12] propose a cross-domain classification method. Starting from a set of labeled data in a source domain and determining domain-independent words (features) that occur both in the source and the target domain, they construct a feature bipartite graph that models the relationship between domain-specific words and independent words. To obtain the domain specific words they use an adapted spectral clustering algorithm on the feature graph

Based on these premises, we propose a method to construct an opinion vocabulary by expanding a small set of initial (seed) words with the aid of connectives. The method consists of four steps, as follows:

1. Given a positive word seed list and a negative word seed list and making use of WordNet's synsets, we expand the initial seed lists based on the synonymity / antonymy relations.

The initial words will be assigned a score of 1 for positive words and -1 for negative words, respectively. We compute the orientation score for each newly found word by recursively processing the synsets for each seed word. A word can be found in synsets corresponding to different seed words, either in a synonymity or antonymy relations. Another factor we take into account is the *distance* between the seed word and the currently processed word, as provided by the WordNet hierarchy. From these two considerations, a more formal way to compute the score of a word (s_w) to be added to the seed list is:

$$s_w = \max(abs(s_{w,o}) \cdot sign(\max(s_{w,o})))$$

where

$$s_{w,o} = \begin{cases} f \cdot s_o, when\ w\ and\ o\ are\ synonyms \\ -f \cdot s_o, when\ w\ and\ o\ are\ antonyms \end{cases}$$

and o is a seed word, while f is a parameter for which we empirically assigned values between 0 and 1 (in our current implementation $f = 0.9$); in our future work we plan to determine its value by optimization.

The result of this step is an expanded seed word list together with their orientation score.

2. From a corpus of documents, we parse and extract all adjectives and conjunctions, constructing a set of relationships between the determined words. There can be two types of relationships, indicating if two or more words have the same context orientation (words connected by *and*, *or*, *nor*) or opposite orientation (words connected by *but*, *yet*). We will refer to them in the following algorithms as *ContextSame* and *ContextOpposite* relations, respectively.

```
G = ({}, {})
foreach document d in corpus
 foreach sentence s in d
     parseTree = GetParseTree(s)
     {w,c} = RetrieveWordsAndConjunctions(parseTree)
     ConstructRelationGraph(G, {w, c})
     HandleNegation(G, s)
```

Figure 1. The algorithm for constructing the relationship graph G.

Based on the determined relations, we can then construct a relationship graph $G(W, E)$, where

- W=*{set of determined adjectives}* and
- $E=\{w_i w_j$, where w_i, w_j from W if there is a determined relationship between w_i and w_j, each edge having a positive weight for the *ContextSame* relationship and a negative weight for the *ContextOpposite* relationship}.

In what follows, we describe the algorithm for building the relationship graph G (see Figure 1).

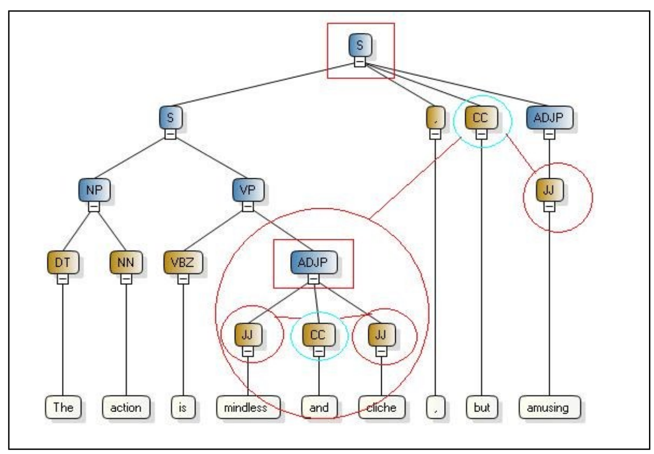

Figure 2. The parse tree and analysis of the sentence "The action is mindless and cliché, but amusing". We identify *mindless, cliché, amusing* as adjectives (having the JJ tags) connected by *and*, *but* (having the CC tags).

We used a maximum entropy parser[1] to retrieve a sentence's parse tree that we then analyze in the *RetrieveWordsAndConjunctions* procedure. We construct an adjective stack w and a conjunction stack c by extracting the relevant nodes according to their part-of-speech tags and group them together based on the common parent node between the adjective nodes and the conjunctions nodes. In the *ConstructRelationGraph*, we will add the nodes for each newly found adjective and add new edges to the relationship

[1] http://sharpnlp.codeplex.com/

graph G according to each conjunction's behavior. Each edge has an associated weight with values between 0 and 1, determined by optimization. We handle the presence of negation in the sentence by reversing the type of the relation, if a negation is detected. For example, considering the sentence "*Some of the characters are fictitious, but not grotesque*", the initial relation between *fictitious* and *grotesque* would be a *ContextOpposite* relationship, but the presence of the negation is converting it to a *ContextSame* relationship. We depict another example visually, in Figure 2.

3. The third step implies cleaning the resulting set of words and relationship graph by removing stop words and self-reference relations. Consider the example "*The movie has a good casting and a good plot*". The algorithm detects a *ContextSame* relationship between the adjective *good* and itself. Since there is no useful context information we can use, we do not want them to influence the results of the scoring done in the next step.

4. In the fourth step, we determine the orientation of the words extracted from the corpus by applying an algorithm on the relationship graph obtained in the previous steps, which was inspired by the well-known PageRank algorithm [2]. For this, we define two score vectors, a positivity score *sPos* and a negativity score *sNeg*, respectively. We choose the final score to be the sum of the positivity and negativity score. The sign of the score represents the word's orientation, that is, a positive score characterizes a positive opinion orientation, while a negative score characterizes a negative opinion orientation. The algorithm is presented in Figure 3, and described in what follows.

```
InitializeScoreVectors(sPos(W), sNeg(W))
do {
foreach word w_i in W
    foreach relation rel_ij in relationship graph G that contains w_i
        if rel_ij is a ContextSame relation
            sPos(w_i) += weigth(rel_ij) * prevSPos(w_j)
            sNeg (w_i) += weigth(rel_ij) * prevSNeg(w_j)
        else if rel_ij is a ContextOpposite relation
            sPos(w_i) += weigth(rel_ij) * prevSNeg(w_j)
            sNeg (w_i) += weigth(rel_ij) * prevSPos(w_j)
    NormalizeScores(sPos(w_i), sNeg(w_i))
} while more than 1% of the words w_i in W change orientation
```

Figure 3. The algorithm for determining the orientation of words extracted from a corpus.

We initialize the score vectors based on the orientation scores of the expanded seed word list (see step 1). We will assign the corresponding positivity or negativity score sw_j for each adjective w found in the seed list. For the opposite score we assign a very small value (ε), in order to allow for meaningful values when computing the score for *ContextOpposite* relations.

A *ContextSame* relation enforces the existing positive and negative scoring of w_i proportionally with the scoring of w_j. A *ContextOpposite* enforces the negativity score of w_j with respect to the positivity of w_i, and the positivity score of w_j with respect to the negativity score of w_i.

3. USE CASE: TWITTER MOVIE COMMENTS

Concerning the movie domain, research was done in classifying movie reviews by overall document sentiment [8], but there are few lines of research connecting the movie domain with social media. Sitaram Asur and Bernardo A. Huberman [1] demonstrate how sentiments extracted from Twitter can be used to build a prediction model for box-office revenue.

Our aim is to see how well a domain specific vocabulary constructed from movie reviews performs when applied to analyzing tweets. We used a document corpus of 27,886 IMDb (Internet Movie Database) movie reviews [3] and constructed a movie domain specific vocabulary according to the approach presented in Section 2. We retrieved 9,318 words, from which 4,925 have a negative orientation and 4393 have a positive orientation. Table 1 shows a few examples of positive and negative adjectives extracted from the movie review corpus.

Table 1. Examples of adjectives that were extracted.

Positive words	Negative words
surprised, original, breathless, chilling, undeniable, disturbing, irresistible, speechless, stylized, amazed, provoking, shocking, undisputed, unforgettable, electrifying, enraptured, explosive, unanticipated, unforeseen, recommended	syrupy, uninspiring, forgettable, frustrating, mild, contrived, laughable, restrained, showy, preachy, amateur, dogmatic, edgeless, foreseeable, ordinary, standard, saleable, usual, predictable

Table 2. Top opinion words identified for the highest and lowest ranking movies in our search

Inception (2010)	Meet the Spartans (2008)
Positive words: good, great, awesome, amazing, favorite, fantastic, incredible, thrilling, different, speechless *Negative words*: bad, confusing, weird, stupid, dumb, boring, predictable, horrible, disappointing	*Positive words*: funny, awesome, great *Negative words*: bad, stupid, dumb, weird, silly, common, ridiculous, terrible

For our tests, we crawled 220,387 tweets, using the Twitter Search API[6], over a two month interval, keyed on 84 movies, spanning different genres and release dates. As search keywords we used the movie name and the *movie* tag, in order to increase the relevance of the results. We used a simple tokenizer to split the text of the retrieved tweets and kept the tokens that had a dictionary entry as adjectives. We then matched the tweet adjectives to our domain specific vocabulary. For all subsequent analysis we only considered adjectives that were used in tweets and also appeared in our vocabulary, since we were intersected to see the relevance of our vocabulary in terms of actual usage and frequency over time. Without actually classifying each tweet, we counted the frequency of positive and negative opinion words that we identified in the collection of tweets. An example of top opinion words that we identified for the highest and lowest

[3] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[6] http://search.twitter.com/api/

ranking movies are shown in Table 2. Table 3 presents a sample of the movies that we analyzed, showing for each movie the genre, number of tweets, our score obtained by counting the positive opinion words and the IMDb score. In Figure 4 we represent graphically the positive and negative opinion word counts for the movie *Inception*.

Table 3. A sample of the movies that we analyzed, showing for each movie the genre, number of tweets, our score obtained by counting the positive opinion words and the IMDb score.

Movie	Genre	Our score	IMDb score	Tweets
Inception (2010)	mystery, sci-fi, thriller	66.52	8.9	19,256
Megamind (2010)	animation, comedy, family	67.71	7.3	8,109
Unstoppable (2010)	drama, thriller	63.67	7	15,349
Burlesque (2010)	drama, music, romance	70.78	6.2	1,244
Meet the Spartans (2008)	comedy, war	40.67	2.5	44
Pootie Tang (2001)	comedy, musical	45.88	4.5	79
Matrix (1999)	action, sci-fi	56.65	8.7	1,947
Blade Runner (1982)	drama, sci-fi, thriller	56.65	8.3	407
Metropolis (1927)	sci-fi	66.23	8.4	419

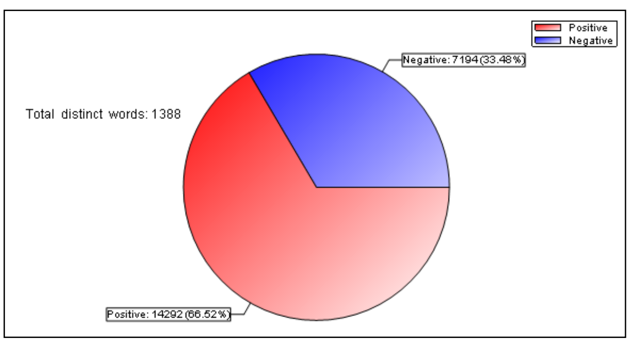

Figure 4. Word distribution for the movie *Inception* over 19,256 tweets.

In the cases presented in Table 3, there is a relationship between the number of positive opinion words and the rating from IMDb. One thing to notice is that in IMDb the movie ratings can be roughly grouped in three categories: ratings between seven and ten points accounting for good and very good movies, between five and seven points for average movies, and below five points for poor quality movies. Our positive opinion word count has a maximum of approximately 70 (or seven on a scale from zero to ten). In our future work we plan to conduct a series of experiments in order to determine if there exists a correlation between the two numbers: the IMDb rating and the number of positive opinion words. This involves collecting a higher number of movie related tweets (in the order of hundreds) in order to be able to report significant results.

4. CONCLUSION AND FUTURE WORK

In this paper, we presented an approach to identifying opinion diversity expressed within text, with the aid of a domain-specific vocabulary. As a use case, we processed a corpus of IMDb movie reviews, extracted a set of adjectives together with their opinion orientation and used the generated opinion lexicon to analyze a different opinion source corpus, i.e. a tweet collection. For future work, we plan to further extend our algorithm to include opinion words expressed by verbs and adverbs, as well as more complex expressions. A second item point is carrying out a set of experiments in order to determine the correlation between positive opinion words for a given movie and the IMDb movie rating. Thirdly, from the lessons learned, we would look into applications in other domains like product reviews.

5. ACKNOWLEDGMENTS

The research leading to these results has received funding from the Slovenian Research Agency and the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°257790.

6. REFERENCES

[1] Asur, S. and Huberman, B. A. 2010. *Predicting the Future With Social Media*. In Proceedings of the ACM International Conference on Web Intelligence.

[2] Brin, S. and Page, M. 1998. *Anatomy of a large-scale hypertextual Web search engine*. In Proceedings of the 7th Conference on World Wide Web (WWW).

[3] Esuli, A. and Sebastiani, F. 2006. *SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining*. In Proceedings of the 5th LREC.

[4] Fellbaum, Ch. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

[5] Gamon, M and Aue, A. 2005. *Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms*. In Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP.

[6] Hatzivassiloglou, V. and McKeown, K. 1997. *Predicting the semantic orientation of adjectives*. In Proceedings of the 35th Annual Meeting of the ACL.

[7] Kim, S-M. and Hovy, E. 2004. *Determining the sentiment of opinions*. In Proceedings of COLING.

[8] Pang, B. and Lee, L. 2002. *Thumbs up? Sentiment Classification using Machine Learning Techniques*. In Proceedings of EMNLP.

[9] Turney, P. D. 2002. *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*. In Proceedings of the 40th Annual Meeting on ACL.

[10] Jijkoun, V., de Rijke, M. and Weerkamp, W. 2010. *Generating Focused Topic-Specific Sentiment Lexicons*. In Proceedings of the 48th Annual Meeting of the ACL.

[11] Kanayama, H. and Nasukawa, T. 2006. *Fully Automatic Lexicon Expansion for Domain-Oriented Sentiment Analysis*. In Proceedings of the EMNLP.

[12] Pan, S. J., Ni, X., Sun, J-T, Yang, Q. and Chen, Z. 2010. *Cross-Domain Sentiment Classification via Spectral Feature Alignment*. In Proceedings of the World Wide Web Conference (WWW).