Disambiguating Context-Dependent Polarity of Words: an Information Retrieval Approach

Olga Vechtomova
Department of Management Sciences
University of Waterloo
200 University Ave. W., Waterloo, ON, N2L 3G1, Canada
ovechtom@uwaterloo.ca

Abstract

The paper introduces $PolaritySim$ – a novel approach to disambiguating context-dependent sentiment polarity of words. The task of resolving the polarity of a given word instance as positive or negative is addressed as an information retrieval problem. At the pre-processing stage, a vector of context features is built for each word w based on all its occurrences in the positive polarity corpus (consumer reviews with high ratings) and another vector – on its contexts in the negative polarity corpus (reviews with low ratings). Lexico-syntactic context features are automatically generated from dependency parse graphs of the sentences containing the word. These two vectors are treated as "documents", one with positive and one with negative polarity. To resolve the contextual polarity of a specific instance of the word w in a given sentence, its context feature vector is built in the same way, and is treated as the "query". An information retrieval (IR) model is then applied to calculate the similarity of the "query" to each of the two "documents", with the polarity of the best matching "document" attributed to the "query". The method uses no prior polarity sentiment lexicons or purposefully annotated training datasets. The only external resource used is a readily available corpus of user-rated reviews. Evaluation on different domains shows more effective performance compared to state-of-the-art baselines, Support Vector Machines (SVM) and Multinomial Naive Bayes (MNB) classifiers, on three out of four datasets. $PolaritySim$, SVM and MNB were also evaluated with an out-of-domain training corpus. The results indicate that $PolaritySim$ is more effective and robust when used with an out-of-domain corpus compared to SVM and MNB. We conclude that an IR based approach can be an effective and robust alternative to machine learning approaches for disambiguating word-level polarity using either within-domain, or out-of-domain training corpora.

Keywords: sentiment analysis; polarity disambiguation; word polarity; context-dependent polarity of words.

1 Introduction

The popularity of online review sites has led to an abundance of content written by consumers. For example, a recently released Amazon corpus (McAuley et al., 2015) contains 142.8 million reviews across a wide range of categories covering the period from 1996 to 2014. Most consumer reviews have overall ratings, representing the reviewer's satisfaction with the product or service. Rated reviews are readily available sources of rich contextual information representing how words are used in positive and negative contexts. We propose $PolaritySim$ – an extensible method for identifying the context-dependent polarity of words expressing an opinion about another word or phrase (opinion target). The only external resource required is a review corpus with user-assigned numerical ratings. The method determines sentiment valence of words with ambiguous (e.g. "small") or unambiguous (e.g. "beautiful") sentiment, as well as words that do not carry sentiment valence on their own, but acquire it through context. For example, it correctly determines the negative polarity of "eat" in "This camcorder eats up tape". The task of disambiguating the polarity of a given word instance as positive or negative is addressed as an

information retrieval (IR) problem. At the pre-processing stage, we build one vector of all contexts of the word *w* in the positive set (i.e. reviews with high ratings) and another vector – of its contexts in the negative set (reviews with low ratings). The lexico-syntactic context features are automatically generated from the dependency parse graphs of all the sentences containing the word *w* in the positive or the negative corpus. The resulting positive and negative vectors are treated as "documents". At run time, to determine the polarity of a specific instance of *w* in an unlabeled review, a context vector is built, which is treated as the "query". The context features for this vector are derived only from the current sentence containing this instance of *w*. An information retrieval (IR) model is then applied to calculate the similarity of the "query" to each of the two "documents".

The $PolaritySim$ method is extensible in a number of ways. For example, the words in the context features could be expanded with related words or the feature set can be expanded with co-occurring patterns from adjacent sentences. Section 4.3 describes one such extension, whereby words in the context features are expanded with related words generated using a Word2Vec model.

The rest of the paper is organized as follows: Section 2 outlines the motivations and contributions of this work, Section 3 discusses related work, Section 4 presents the method, Section 5 describes the datasets and evaluation experiments, Section 6 contains the analysis of results, and Section 7 concludes the paper and suggests future research directions.

2 Motivation and contributions of the work

Most research efforts in the sentiment analysis field have been directed at identifying sentiment and its polarity at the sentence or document level. Two major sentiment analysis approaches to date have been: (a) lexicon-based and (b) machine learning based. In the first approach, the polarity of individual words is first determined by using a prior polarity lexicon, then possible polarity shifters are identified, usually by applying hand-crafted rules. Sentence or document level polarities are then calculated by using word counting methods. In the second approach, the machine learning models have to be trained on the training datasets manually labeled at the same level of granularity (phrase, sentence or document) as the test dataset. The main limitation of these approaches is their reliance on external resources, such as lexicons in the lexicon-based approaches and purposefully-built training datasets in the machine-learning based methods, which typically require substantial human effort to construct.

The main objective of this research is to develop a method for disambiguating contextual sentiment polarity at the lowest level of granularity – words, without relying on any purposefully-built training datasets and lexicons. Polarities of individual words are highly dependent on their context. Among the factors that can affect word polarity are: the target of opinion, e.g."long rebooting time" (negative) vs. "long battery life" (positive), whether the word is used ironically or sarcastically, presence of phrases intensifying, reversing or diminishing the polarity of the word (e.g. "never", "too", "barely", "hardly", "even"). Instead of relying on prior lexicons, manually labeled datasets or a large number of handcrafted rules to capture different kinds of polarity shifters, the proposed method determines contextual polarities by using an IR approach and a large body of readily available user-rated reviews. It, therefore, eliminates the need for the manual effort required to build lexicons or datasets.

The method can be readily applied to different categories of user reviews, due to the availability of large datasets of user-rated reviews. It can also be applied to the categories and domains for which no user-rated reviews exist by using out-of-domain reference corpora.

The main theoretical contribution of this research is demonstrating that an IR approach to determining word-level contextual polarity can achieve performance that is comparable to or better than the performance of the state-of-the-art machine learning approaches. The paper also shows that the proposed approach is more robust with out-of-domain training corpora than the state-of-the-art machine learning approaches.

A number of practical applications can benefit from knowing word-level contextual polarity, such as generating text with custom recommendations for users based on existing reviews, extraction of specific positive and negative expressions referring to an entity, multi-document summarization of reviews, as

well as question answering and information retrieval for complex information needs. For example, if a user has the information need: "Find a camera that works well in poor lighting conditions", it would be useful for the system to know the contextual polarities of "sharp" and "low" in the sentence "The picture was sharp, even in low light.", so that it can determine whether it is relevant to the user's information need.

The specific contributions of this work are summarized below:

- The proposed method uses reference corpora with document-level positive and negative polarity labels to disambiguate context-dependent polarity of individual words in an unlabeled document;
- Readily available user reviews with overall numerical ratings are demonstrated to be effective positive and negative reference corpora for determining word-level contextual polarity;
- The method is compared to state-of-the-art baselines and proves to be more effective on three out of four datasets, achieving accuracy in the 83%-91% range in different subject domains using within-domain reference corpora;
- The method is more effective and robust than machine learning approaches with out-of-domain reference corpora, achieving performances at or above 80%.
- An information retrieval based approach is shown to be a state-of-the-art alternative to machine learning approaches for determining word-level contextual polarity;
- Lexico-syntactic features are more effective than lexical or syntactic features alone;
- Four new datasets were developed for evaluating word-level contextual polarity disambiguation methods, and are available for academic research.

3 Related Work

Sentiment analysis has received considerable attention over the past fifteen years. The body of research in this field can be grouped into three categories based on the linguistic units for which sentiment is predicted: words/phrases, sentences and documents. The majority of research effort has been focused on detecting sentence- and document-level sentiment and its polarity. There exist a number of comprehensive surveys that summarize and describe approaches in each of the three categories (Mohammad, 2015; Liu, 2015). Word-level sentiment polarity research can be grouped into three areas:

- Sentiment lexicons with prior polarities;
- Contextual polarity;
- Target-based sentiment polarity.

Since our approach is aimed at identifying polarity at the word level, we focus on reviewing research in this category. A number of relevant works aiming to resolve polarity at the sentence- or document-level, which address contextual word polarity as part of their methods are also reviewed.

3.1 Sentiment polarity lexicons

Most approaches towards generating polarity lexicons assume that each word has a prior polarity, i.e. it has the same sentiment in the majority of its usages. For example, words "beautiful" and "exquisite" are mostly used in positive contexts, while "frustrating" and "bland" in negative. Prior polarities are also referred to as prior associations (Mohammad, 2015).

Some sentiment lexicons have been built manually, e.g. the Affective Norms for English Words (ANEW) has manual valence labels for 1034 words (Bradley et al., 1999), AFINN (Nielsen, 2011) has valence labels for 2477 words. One of the most widely used sentiment lexicons is the Multi-Perspective

Question Answering (MPQA) lexicon (Wilson et al., 2005), which contains 8222 words from manually labeled resources and automatically built lists based on annotated and unannotated data. The words are labeled with sentiment strength (strong, weak) and prior polarity (positive, negative). Some recent works used crowdsourcing to generate lexicons, e.g. the lexicon created by Warriner et al. (2013) contains valence for 13,915 words. A number of researchers approached the construction of lexicons semi-automatically or automatically, e.g. (Hatzivassiloglou and McKeown, 1997; Turney and Littman, 2003; Hu and Liu, 2004; Mohammad et al., 2013; Severyn and Moschitti, 2015). Hu and Liu (2004) relied on WordNet synonyms of words with known polarity to infer the polarity of other words. Turney and Littman (2003) classified words into positive and negative based on their co-occurrence with words known to have positive or negative association. Mohammad et al. (2013) used the same approach as Turney and Littman (2003) to generate a sentiment lexicon from tweets. Esuli and Sebastiani (2006) created a sentiment resource called SentiWordNet, where each WordNet synset has a score associated with positive, negative and objective valence. The resource was constructed in a recursive semi-supervised manner. The process started with a small set of seed words, which were then recursively expanded with synonyms and antonyms in WordNet. The polarity of the original seed word was attributed to its synonyms, and reversed for its antonyms. The latest version, SentiWordNet 3.0 is described in (Baccianella et al., 2010). Tang et al. (2014) proposed a method for building large sentiment lexicons from tweets by using a neural network approach. As input data, they used a combination of seed words and emoticons in tweets. Chetviorkin and Loukachevitch (2014) built polarity lexicons from tweets using co-occurrence statistics and Markov random field framework. Severyn and Moschitti (2015) developed an automatic method to generate sentiment lexicon from tweets using distant supervision. Their method relies on cues, such as positive and negative emoticons, to infer the overall sentiment expressed by a tweet. They then train an SVM model with features derived from unigrams and bigrams extracted from tweets to predict the sentiment association score for each lexicon entry.

3.2 Contextual polarity

While some words taken out of context have a stronger prior association with a specific polarity, their individual instances may convey the opposite polarity. Furthermore, words that usually have no sentiment connotation may acquire it in a specific context of use. For this reason, the use of sentiment polarity lexicons alone is not sufficient to determine polarity of specific instances of words in context. Contextual polarity can be dependent on the following factors:

- The target of the sentiment. Consider as an example, the following two usages of "cold": "The coffee was cold." and "The lemonade was cold." The word "cold" is weakly subjective with the negative prior polarity in the MPQA lexicon. However, in the absence of other contextual clues, most readers will likely view the first example as negative and the second as positive, since based on our common knowledge, we expect coffee to be hot, while lemonade – cold.

- Other linguistic clues in the same sentence or the rest of the document that either reinforce, weaken or reverse the prior polarity. An example of a sentence where prior polarity is reinforced is "The coffee was nice and strong, just as I hoped." The word "strong" is weakly subjective with the positive prior polarity in the MPQA lexicon. This sentence contains several clues reinforcing the prior polarity: (1) conjunct "and" relationship with a strongly subjective positive polarity noun "nice"; (2) clause "just as I hoped", which indicates that the person's expectations were met. An example of prior polarity weakening expression is "The coffee was not as strong as I hoped it would be." and polarity reversal: "The coffee was not strong at all and had a bland taste." Polanyi and Zaenen (2006) give a comprehensive overview of different factors in language that can affect valence.

There have been a number of approaches to identify contextual polarity shifters either using handcrafted rules or machine-learning approaches. Polarity shift techniques are commonly used in conjunction with prior polarity lexicons, whereby the prior word polarity is taken from the lexicon, which is then

either confirmed or shifted based on the polarity shift methods. Wilson et al. (2005) proposed such an approach, where the method starts by identifying expressions containing sentiment clues from a sentiment lexicon with 8000 entries. The goal of the second stage is to identify contextual polarity of such expressions using a machine learning approach. The features used by the classifier include bag-of-word features surrounding the expression, modification features (e.g. preceded by an adjective), structure features derived from the dependency parse tree (e.g. whether there is "subj" relationship in the path towards the root), sentence features (e.g. subjective clues in previous sentences), and a document feature representing the topic of the document. Kennedy and Inkpen (2006) used a sentiment lexicon with negation, intensifier and diminisher terms in both term-counting method and a machine-learning method for document-level polarity prediction. The use of valence shifters in both methods improved performance. Ding et al. (2008) used an opinion lexicon and a large set of rules, such as negation rules, intra- and inter-sentence conjunction rules, synonym and antonym rules, to identify polarity expressed with respect to product features in consumer reviews. Ikeda et al. (2008) proposed a model which identifies cases where the prior polarity of a sentiment word is different from the polarity of the sentence it occurs in. Features generated from such polarity-shifted words are used in a machine learning model to classify polarity of sentences. Joshi and Penstein-Rose (2009) used dependency relation triples where one of the words has been "backed off" to its part-of-speech as features in machine learning based sentence polarity classification. Kessler and Schuetze (2012) proposed a supervised method for determining polarity of sentences containing inconsistent sentiment words, i.e. words that shift polarity valence depending on the context. They address the task of sentence-level polarity classification by using a machine learning method trained on a corpus annotated with sentence-level polarity. One of the novel contributions of their method is automatic extraction of polarity reversing syntactic structures as features. Li et al. (2013) identified five categories of polarity shifting structures: negation, contrastive transition, modality, implication and irrelevance. They addressed them using a rule-based approach, for example, if a negation trigger word is found in the same clause as the sentiment word, its polarity is reversed. The method was evaluated on the document-level sentiment classification task. Xia et al. (2016) proposed a multi-stage approach to document-level polarity prediction, which includes rule-based methods for identifying negation and contrast, as well as a method to re-write a negated phrase with a non-negated one, e.g. "I don't like the movie" to "I dislike the movie" using an automatically built antonym dictionary. The rationale is that this transformation may eliminate errors caused by negation in the bag-of-words machine-learning approaches.

The main limitations of lexicon-based approaches are threefold. Firstly, they rely on prior polarity lexicons, which may not contain the word in question. Secondly, the word's prior polarity with respect to the given target may be different from the prior polarity recorded in the lexicon. Thirdly, the polarity reversal rules can capture some syntactic constructs, e.g. negation, modal verbs, but not other more complex clues, such as "just as I hoped" clause in an earlier example. The main limitation of machine-learning based methods is that they typically require training datasets manually annotated at the phrase or sentence level.

Some approaches are aimed at building context-dependent sentiment lexicons (Fahrni and Klenner, 2008; Wu and Wen, 2010; Lu et al., 2011; Weichselbraun et al., 2013) or domain-specific sentiment lexicons (Lau et al., 2011). Fahrni and Klenner (2008) used conjunctions of ambiguous adjectives with unambiguous ones with known polarity from an opinion lexicon, and also extracted groups of related target words from Wikipedia in order to build a target-specific adjective polarity lexicon. Wu and Wen (2010) proposed a method for building a lexicon with the so called "semantic expectations" of nouns (e.g. "price" has a negative semantic expectation, but "salary" has a positive expectation). They mine the web with a lexico-syntactic pattern (e.g. "[noun] is a little [adjective]") and use the number of hits to infer polarity expectation of a specific noun, e.g. "price is a little low" has more hits than "price is a little high", therefore "price" has negative polarity expectation. Lu et al (2011) propose an optimization approach for building a domain-specific contextual polarity lexicon consisting of aspect–sentiment word pairs, such as "screen – big; polarity=positive". They use a number of resources, such as general-purpose sentiment

lexicons, consumer-rated product reviews, thesauri, such as WordNet and a number of linguistic heuristics, e.g conjunction with other words of known polarity, "but" clauses and negation rules. Brun (2012) extracted noun-adjective patterns from product reviews using a syntactic parser, frequencies of occurrence in positive and negative product reviews and hand-crafted rules. A clustering method was then used to group them by polarity. Weichselbraun and Gindl (2013) start by identifying ambiguous words using a training dataset of consumer-rated product reviews. Statistical methods are then used to collect context terms from positive and negative documents to build a contextualized sentiment lexicon. Both unambiguous words from a prior polarity lexicon and ambiguous words with their contextual polarity are then used in calculating document-level polarity. In (Lau et al., 2011) a domain-independent polarity lexicon is expanded using a co-occurrence measure to obtain a domain-specific lexicon, and the sentiment words are weighted based on their probability of occurrence in the known positive and negative documents. The method was evaluated on the document level polarity classification task in a number of domains, including the finance domain, for which no labeled training datasets are available. This method was extended in (Lau et al., 2012) to associate sentiment words with "aspects", noun phrases located in proximity of a sentiment word occurrence. The above methods determine contextual polarity of an ambiguous word at the domain level. In (Fahrni and Klenner, 2008; Wu and Wen, 2010; Lu et al., 2011; Brun, 2012) one domain-level polarity is attributed to a target-sentiment word pair, while in (Weichselbraun et al., 2013; Lau et al., 2011) one domain-level polarity is attributed to a sentiment word. In contrast, the goal of our method is to determine contextual polarity of the *specific* instance of a word in a given sentence, since there can be multiple polarities per word or even per word-target pair in a domain. For example, while "dry" in "The salad leaves were dry and crunchy" is positive, it is negative in: "The salad leaves were wilted and dry". Furthermore, our method implicitly accounts for any other lexico-syntactic clues from the specific context of the given word that may affect its polarity.

Socher et al. (2013) used crowdsourcing to develop Sentiment Treebank, which contains sentiment polarity association values assigned to nodes in a syntactic parse tree. The resource contains 9645 sentences from the movie review corpus created by Pang and Lee (2005). The resource was built by parsing each sentence in the corpus, and giving the text string corresponding to each node in the parse tree in a random order to annotators. The annotators were asked to assign a score to each string on a 25-point scale from very negative to very positive. Each string is annotated by three annotators, whose scores are averaged to produce the final polarity score. In contrast to some previous approaches, e.g. (Lu et al., 2011), each word has only one polarity score in the Sentiment Treebank, so context-specific polarity can be obtained from the higher-level nodes in the parse tree, e.g. "bad" has a corpus-wide score of 4, but a higher-level node in the parse tree subsuming it may have a different score, e.g. 18 for "it's not too bad". Sentiment Treebank is used to train a deep learning model called Recursive Neural Tensor Network (Socher et al., 2013).

Wang and Manning (2012) conducted a study comparing a number of state-of-the-art methods for determining polarity of snippets and full-length product reviews, including methods using lexicons and polarity reversal rules, as well as models learned from parse trees (Nakagawa et al., 2010; Socher et al., 2011) to Support Vector Machines (SVM) and Multinomial Naive Bayes (MNB) classifiers with unigrams and/or bigrams as features. The results show that SVM and MNB perform better than these state-of-the-art methods on a range of datasets. MNB showed better performance than SVM on snippets, while SVM performed better with full-length reviews.

3.3 Target-based sentiment polarity detection

Work in this category is aimed at determining sentiment polarity towards a given target. The target can be a named entity, e.g. a company, product or person name or its specific aspect, such as price or quality when referring to a product. The goal of such methods is therefore not necessarily to determine polarity of individual words, but to identify whether a given sentence or document expresses an overall positive or negative polarity towards the target entity or its aspect. Jiang et al. (2011) developed a method for identifying polarity towards a given entity in tweets. They used a machine learning approach with a

variety of features, including target-independent features, such as emoticons and hashtags, and target-dependent features, generated from a syntactic parse tree, each representing specific type of syntactic relationship that a word may have with the target. While we also use syntactic information, our approach is fundamentally different from theirs. First, their approach is machine learning based and requires tweets manually annotated with target-based polarity. In contrast, our method is IR based and requires no target-level annotations; secondly they use hand-crafted rules for generating syntactic features, whereas we use all syntactic relations with the target word within the distance of three nodes in the syntactic dependency parse tree. The most important difference however, is that their method does not determine polarity at the word level, so if a sentence contains mixed polarity, such as "The camera's battery is bulky, but it lasts a long time", their method only outputs one polarity label.

Research on target-based sentiment has recently been facilitated by the Aspect-Based Sentiment Analysis (ABSA) shared task (Pontiki et al., 2015) in SemEval 2014, 2015 and 2016. One of the subtasks required participating systems to determine whether a given target, called aspect, in the sentence has a positive, negative or neutral opinion directed at it. Most top-performing participating systems used machine learning approaches, relying on hand-labeled training dataset and/or manually constructed sentiment lexicons. For example, one of the top-performing systems in SemEval ABSA 2015 (Zhang and Lan, 2015) used SVM with a variety of features, including sentiment lexicon features derived from MPQA and SentiWordNet, linguistic features including n-grams, POS tags, grammatical relationships, and other features such as domain-specific word lists. The top performing system in 2016 (Brun et al., 2016) used a machine learning approach with a number of features associated with each term, including the semantic class generalizing the meaning of the term (e.g. food, service), bigrams and trigrams including this term and all syntactic dependencies of this term. Both of the above methods were trained on ABSA training datasets, containing the aspects found in each sentence and the polarity of the sentiment expressed by them.

Target-based polarity methods, such as the ones developed for the ABSA shared task are not aimed at identifying contextual polarity of individual words in a sentence, but rather polarity or polarities expressed in the entire sentence towards a specific target. The approach proposed in this paper aims to determine polarity at the word level.

4 Methodology

The overall system architecture is presented in Figure 1, and the detailed description of each stage is given in the following sections. In Stage 1 (Section 4.1) the system pre-processes the positive and negative corpora to generate a positive (*posV*) and negative (*negV*) vectors of context features for each word. In Stage 2 (Section 4.1), the system is given a sentence from an unlabeled document, and for each word instance, it builds a context feature vector (*EvalV*) using only the content of this sentence. Then, in Stage 3 (Section 4.2) the system computes pairwise similarity of the word's vector in the given sentence (*EvalV*) to the positive (*posV*) and negative (*negV*) vectors of the same word generated from the positive/negative corpora in Stage 1. The polarity is assigned to this instance of the word depending on whether its *EvalV* is more similar to the positive (*posV*) or the negative (*negV*) vectors.

4.1 Context feature vector construction

The following steps are performed on each of the two reference corpora: positive and negative. Each sentence in a positive/negative corpus is processed by using the Stanford CoreNLP dependency parser (Manning et al., 2014). In each sentence, we first locate all nouns or personal pronouns (*n*). These are the potential opinion targets. Then, for each *n*, its dependency triples with all adjectives, nouns and verbs (*w*) are extracted, where the dependency relation is either an adjectival modifier (amod), nominal subject (nsubj), passive nominal subject (nsubjpass), direct object (dobj) or relative clause modifier (rcmod). Figure 2 shows a dependency parse graph. An example of dependency triple from this sentence is nsubj(soft, bagels), where nsubj (nominal subject) is a syntactic relationship, "bagels" is a governor, while "soft" is a dependent word. We also identify dependency relations of adjectival complements

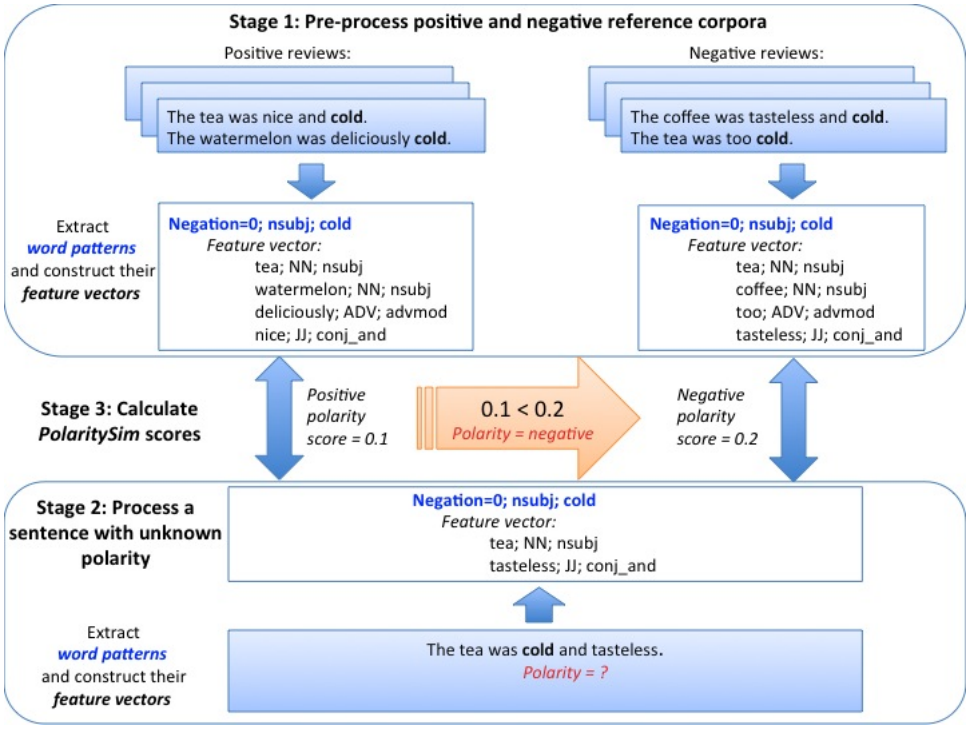


Figure 1: System architecture

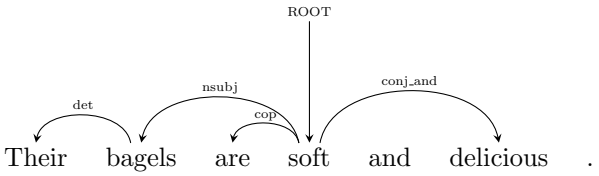


Figure 2: Dependency parse (example 1)

(acomp) and open clausal complements (xcomp), and merge them with the nominal subject relationship sharing the same verb, e.g. "nsubj(menu, looks)" and "acomp(looks, great)" extracted from the sentence "The menu looks great" are merged into "nsubj_acomp(menu, look_great)".

A small set of rules was defined to determine whether the context containing an instance of *w* is negated or not (see Section 4.1.2). For each occurrence of *w* the following information is recorded:

negation (1 – *w* is negated; 0 – *w* is not negated);

dependency relation (DepRel) of *w* with *n*;

lemma of *w* (i.e. the canonical form of *w*);

part-of-speech (POS) of *w*.

These data items form a pattern *p*. For example, "NEGATION=0; nsubj; soft, JJ" is a pattern generated from the sentence in Figure 2. The reason for building vectors for lexico-syntactic patterns as opposed to just words (lemmas), is that, firstly, we want to differentiate between the negated and non-negated instances, and, secondly, between various syntactic usages of the word. For instance, adjectives occurring in a post-modifier position (e.g., in nsubj relationship to the noun) tend to be used more frequently in evaluative manner compared to those used in pre-modifier position (c.f: "tea was cold" and "cold tea"). While "cold tea" usually refers to a type of drink, "tea was cold" has an evaluative connotation. Also, the types of dependency relations they occur in can be different, e.g. adjectives in post-modifier position occur more with certain adverbial modifiers, which can give clues about the adjective's polarity, such as "barely", "too", "overly" or "hardly".

Next, a context feature vector ($posV_p$ from the positive corpus and $negV_p$ from the negative corpus) is built for each *p*, as follows: for each instance of *w* matching this pattern in the corpus (positive or negative, respectively) we extract all dependency relations containing it. Each of them is transformed into a context feature *f* of the form: "lemma; Part-of-Speech (POS); dependency relation". For instance, if adjective "soft" occurs in a dependency triple "conj_and(soft, delicious)", the following feature is created to represent "delicious" and its syntactic role (conjunct) with respect to "soft": "delicious, JJ, conj_and". For each feature *f* we record its frequency of co-occurrence with the pattern *p*. For example, the features for the pattern "NEGATION=0; nsubj; soft, JJ" generated from the sentence in Figure 2 are:

bagel; NNS; nsubj

delicious; JJ; conj_and

be; VBP; cop

More formally, let P be the set of all patterns extracted from the given corpus C (positive or negative). Each pattern p is a 4-tuple $(negation; DepRel, lemma; POS)$. From each corpus C, a non-zero vector $V_p = \{f_1, f_2, f_3, ..., f_n\}$ is generated for each $p \in P$; each f_i corresponds to a 3-tuple $(lemma, POS, DepRel)$, co-occuring in a dependency relation (DepRel) with p in the corpus C. The weight of each feature f_i is set to $freq(p, f_i)$, which is the number of times f_i co-occurs with p in the corpus C. Thus, V_p represents the *global* context of p in C, encompassing all contexts co-occurring with p in the corpus C.

Given a test sentence S, for each pattern p extracted from it, a non-zero vector $evalV_p = \{f_1, f_2, f_3, ..., f_n\}$ is generated, where f_i weight is $freq(p, f_i)$, which is the number of times f_i co-occurs with p in the sentence S.

4.1.1 Composite features

By building features from only directly related dependency triples, we may miss important contextual clues. To avoid this problem, we also build composite features by joining up to three dependency relations as we traverse the dependency graph. By doing so we may generate overly specific composite features that will not match features in the positive/negative reference corpus. Therefore, if the composite feature contains the opinion target word, it is substituted with its part of speech. An example of a composite feature extracted from the sentence: "I love their Italian subs." for the opinion target "subs" and pattern "NEGATION=0; amod; italian; JJ" is "NNS:love:I; NNS:VBP:PRP; amod:dobj:nsubj", where "subs" was substituted with its part-of-speech "NNS". We generate one feature vector for pattern *p* based on its occurrences with all opinion targets. For example, we generate one feature vector for the pattern "NEGATION=0; amod; italian; JJ" from sentences "I liked italian sausage." and "I love their Italian subs." The same method is used to generate a context feature vector ($evalV_p$) for every *p* extracted from the test sentence *s*.

Below are all features generated from relations at distances ≤ 3 in the dependency graph from the sentence: "The pizza was great and the garlic rolls were the best we've had in a while." (Fig. 3) for pattern "NEGATION=0; nsubj; best; JJS":

have:we; VBN:PRP; rcmod:nsubj

have:while; VBN:NN; rcmod:prep_in

be; VBD; cop

great; JJ; conj_and

roll; NNS; nsubj

have; VBN; rcmod

great:be; JJ:VBD; conj_and:cop

great:pizza; JJ:NN; conj_and:nsubj

NNS:garlic; NNS:NN; nsubj:nn

4.1.2 Negation

A set of rules was written to determine whether the text span containing the given word is negated or not. All of the rules use information derived from the lexicalized dependency graph. The reason for

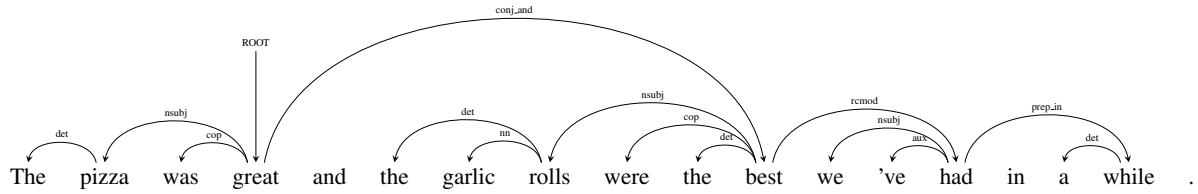


Figure 3: Dependency parse (example 2)

identifying negation is to create a separate context vector for negated and non-negated usages of the word. The rules are summarized below.

1. Check if the verb is negated. For example, in "I don't like their food." nsubj(I, like) and dobj(food, like) are negated. Below are special sub-rules:
   (a) Check if "not" is part of "only" phrase, e.g. "I liked not only their food, but their location and atmosphere." Here, dobj(food, liked), dobj(location, liked) and dobj(atmosphere, liked) are not negated.
   (b) Check for conjunctions that have the same subject, for example, "I don't like or recommend this place." Here, dobj(place, like), dobj(place, recommend), as well as nsubj(I, like) are negated.
   (c) Check if members of the conjunction relation have different subjects. If they do, and if only one of them is negated, then do not apply the negation to the other one. For example, "I don't like the fact that it always restarts at the first song, and I would have preferred to have a random or shuffle mode." There is a conjunction relation between "like" and "preferred", but they each have their own subject, therefore nsubj(I, like) and dobj(fact, like) are negated, but nsubj(I, preferred) and xcomp(have, preferred) are not.

2. Check if the subject noun is negated or if the subject is an indefinite pronoun, such as "nobody" or "nothing". For example, in "No reasonable person will eat here." nsubj(person, eat) and amod(reasonable, person) are negated. In "Nobody will eat this fish." nsubj(nobody, eat) and dobj(fish, eat) are negated.

3. Check if the direct object is negated, e.g. "I had no luck there." Here, nsubj(I, had) and dobj(luck, had) are negated.

4. Process "But" clause negation. This rule checks whether negation expression is in or outside the "but" clause. For example, in "The rice was not salty, but tasty." nsubj(salty, rice) is negated, while nsubj(tasty, rice) is not. In "I liked the rice but not the fish." the dependency triple dobj(liked, fish) is negated, while dobj(liked, rice) is not.

5. Process "neither ... nor" cases. In "The rice was neither salty, nor tasty." both nsubj(rice, salty) and nsubj(rice tasty) are negated. In "I will neither go there again, nor recommend it to others." nsubj(I, go) and dobj(it, recommend) are negated.

6. Process cases with negated direct object. For example, in "It requires no extra cables." nsubj(it, requires), dobj(cables, requires) and amod(extra, cables) are negated. In "I have found not a trace of butter." both nsubj(I, found) and dobj(trace, found) are negated.

These rules capture only the most obvious negation cases, while the more subtle ones can be identified automatically by the proposed method. For example in the sentence "It lacks good screen." the word "lacks" expresses negation, which is not covered by the rules, however it is captured in the vector for the word "good". Assuming that the word "lacks" occurs more in the contexts of "good' in the negative reviews than positive, we will correctly identify the polarity of "good" in this sentence as negative.

4.2 Computing Similarity Between Vectors

We view the problem of computing similarity between vectors as a document retrieval problem. For each pattern *p* found in the test sentence *S*, we compute pairwise similarity between its $EvalV_p$ vector and $posV_p$ and $negV_p$ vectors respectively. The $EvalV_p$ vector is treated as the query, while $posV_p$ and $negV_p$ are treated as documents. Three similarity functions were compared (Table 1): Cosine, TF.IDF and BM25 Query Adjusted Combined Weight (QACW) (Sparck Jones et al., 2000). The BM25 QACW document retrieval model was used as a term-term similarity function for finding related entities (Vechtomova and Robertson, 2012).

Let $V_p = \{f_1, f_2, f_3, ..., f_n\}$ be the non-zero vector for pattern p generated from corpus C (either the positive or negative document set), and $evalV_p = \{f_1, f_2, f_3, ..., f_n\}$ be the non-zero vector for *p* generated from the test sentence S; *F* is the number of features that $EvalV_p$ and V_p have in common; TF_f is the weight of feature *f* in V_p, i.e. the number of times f co-occurs with p in C; QTF_f is the weight of feature *f* in $EvalV_p$, i.e. the number of times f co-occurs with *p* in S; $K = k_1 \cdot ((1-b) + b \cdot DL/AVDL)$; k_1 is the feature frequency normalization factor; *b* is the V_p length normalization factor; *DL* is the number of features in V_p; *AVDL* is the average number of features in vectors for all patterns in the positive/negative set. The *IDF* (Inverse Document Frequency) of the feature *f* is calculated as $IDF_f = log(N/n_f)$, where, n_f is the number of vectors generated from corpus C that contain *f*, and *N* is the total number of vectors generated from corpus C.

Similarity Function	Equation
$PolaritySim_{Cosine}$	$\frac{\sum_{f=1}^{F} QTF_f \cdot TF_f}{\sqrt{\sum_{f=1}^{F} QTF_f^2}\sqrt{\sum_{f=1}^{F} TF_f^2}}$
$PolaritySim_{QACW}$	$\sum_{f=1}^{F} \frac{TF(k_1+1)}{K+TF} \cdot QTF \cdot IDF_f$
$PolaritySim_{TF.IDF}$	$\sum_{f=1}^{F} TF \cdot IDF_f$

Table 1: $PolaritySim$ functions

If $PolaritySim(EvalV_p, posV_p) > PolaritySim(EvalV_p, negV_p)$, positive polarity is assigned to the given instance of *p* in the test sentence, otherwise, negative. The polarity of an instance may be unresolved due to zero similarity between the vector $EvalV_p$ and both vectors $posV_p$ and $negV_p$. In this case, the probability of pattern *p* in each corpus C (positive and negative) is calculated as

$$probability(p) = \frac{freq(p)}{\sum_{i=1}^{P} freq(i)} \quad (1)$$

where $freq(p)$ is the frequency of *p* in the corpus C, P is the total number of patterns extracted from C. If $probability_{pos}(p) > probability_{neg}(p)$, i.e. if the probability of *p* is higher in the positive reference corpus than negative, it is assigned positive polarity, and vice versa.

4.3 Feature expansion with Word2Vec

One possible extension of the model described above is to expand the word in each feature with a set of related words. Word2Vec is a series of shallow neural network models for generating word embeddings (Mikolov et al., 2013). In our work we used the continuous bag-of-words (CBOW) model, trained on reviews with all numerical ratings from the corresponding corpus for each domain (Section 5). The

trained models are then used to obtain a list of words ranked by cosine similarity of their feature vectors to any given word.

Each $EvalV_p$ vector (Section 4.1) only contains features derived from the given sentence. It is, therefore, possible that these features do not exist in $posV_p$ and $negV_p$. For example, consider pattern "NEGATION=0; nsubj; lovely; JJ" from the sentence "The bar area was lovely, cozy, and warm." The features in its $EvalV_p$ vector are:

area; NN; nsubj
cozy; JJ; conj_and
warm; JJ; conj_and
NN:bar; NN:NN; nsubj:nn
be; VBD; cop

The corresponding $posV_p$ vector does not contain the feature "area; NN; nsubj", but contains a feature for the related word "room": "room; NN; nsubj". Since "room" is among the top related words to "area", as output by Word2Vec, we add the matching score for this feature to the *PolaritySim* positive score of this pattern. This extension was evaluated with $PolaritySim_{TF.IDF}$. The matching score $PolaritySim_{W2V}$ is calculated as follows:

$$PolaritySim_{W2V} = PolaritySim_{TF.IDF} + \sum_{f=1}^{F} \sum_{r=1}^{R} TF_r \cdot IDF_r \cdot W2V_r \quad (2)$$

where *r* is a word related to the word in feature *f* through Word2Vec, *R* is the number of top-ranked related words in Word2Vec, and $W2V_r$ is the cosine similarity score assigned to *r* in Word2Vec. We evaluated the following values of *R*: 50, 100 and 500, with 50 showing the best results.

5 Evaluation

The evaluation was conducted on five datasets described in this section. Four datasets (Sections 5.3–5.5) were created by us to evaluate word-level polarity[1]. We also report evaluation on the dataset from the SemEval Aspect-Based Sentiment Analysis (ABSA) shared task in Section 5.6.

5.1 Corpora

Four test datasets described in Sections 5.3–5.5, as well as positive and negative reference corpora were built from three consumer review corpora, described below. The rules for generating positive and negative reference corpora are also described below. These reference corpora were used in constructing $posV$ and $negV$ vectors for the corresponding test dataset.

Restaurant corpus: 157,865 restaurant reviews from one of the major business review websites (Vechtomova et al., 2014). The collection contains reviews for 32,782 restaurants in the United States. The average number of words per review is 64.7. Each review has a consumer-assigned rating in the range of 1-10. Reviews with ratings 1 and 2 are considered negative and constitute the negative reference corpus, while those with rating 10 are considered positive. The corpus contains 63,516 reviews with rating 10, and 18,705 reviews with ratings 1 and 2. Due to this imbalance, we randomly subsampled reviews with rating 10 to match the size of the negative corpus (126,013 sentences).

MP3 corpus: a subset of the Amazon corpus (McAuley et al., 2015), containing 135,943 reviews of products in the category "Consumer Electronics". Each review has a consumer-assigned rating in the range of 1-5. Reviews with ratings 1 and 2 constitute the negative reference corpus, while those with rating 5 are treated as positive. As there is again an imbalance of positive (64,006) and negative reviews (31,792), we randomly subsampled positive reviews to match the size of the negative corpus.

Photography corpus: a subset of the same Amazon corpus containing 273,032 reviews in the category "Photography". Again, we randomly subsampled 150,299 positive reviews to match the size of the negative corpus (42,255 reviews).

[1]The datasets are available for academic research upon request.

5.2 Baselines

As baselines we used Multinomial Naive Bayes (MNB) and linear Support Vector Machines (SVM) classifiers with default settings. Both MNB and linear SVM have been shown to be strong state-of-the-art baselines on sentiment classification tasks, outperforming many rule- and lexicon-based systems (Wang and Manning, 2012). We trained MNB and linear SVM[2] on all sentences from the same positive and negative reference corpora that were used in our methods. As features, we evaluated unigrams ("1-gram" runs in Tables 3−7) and a combination of unigrams and bigrams ("1,2-gram" runs) extracted from each sentence. For MNB ($\alpha = 1$), frequency counts of words in the corpus were used as feature values, whereas for SVM (linear kernel, L2 loss, $C = 1$), their TF.IDF values were used.

5.3 Ambiguous adjectives dataset

This dataset (henceforth referred to as AmbAdj) was designed to evaluate how well methods perform on adjectives that change their valence based on the context. The dataset was constructed from the Restaurant corpus. We chose four measure adjectives (cold, warm, hot and soft) that often refer to food items and may have a positive or negative valence. All reviews with ratings 3-9 were parsed and all dependency triples with one of these adjectives in "nsubj" dependency relation were extracted. The reason why only "nsubj" was used is that post-modifiers are more likely to be used in an opinionated context than pre-modifiers. We extracted adjective instances referring to food names as opinion targets by applying a filter of 456 food names, created by a clustering method (Suleman and Vechtomova, 2016). In total 888 patterns (evaluation cases) were generated as described in Section 4.1, and two annotators were asked to judge them as positive or negative. An example of a test case as it was presented to the annotators is given in Table 2.

Table 2: Example of a test case

Target	Opinion pattern	Original sentence
food	NEGATION=0; nsubj; hot; JJ	The food is always hot and fresh.

The inter-annotator agreement (Cohen's Kappa) is 0.81. The evaluation set consists of 519 cases agreed upon by the annotators. The number of positive/negative cases is 34/180 (cold), 29/25 (warm), 196/10 (hot), and 31/14 (soft). Table 3 summarizes the results.

Table 3: Results on the ambiguous words dataset

Method	Polarity	Precision	Recall	F-measure	Accuracy
MNB (1-gram)	positive negative	0.8716 (258/296) 0.8565 (191/223)	0.8897 (258/290) 0.8341 (191/229)	0.8805 0.8451	0.8651
MNB (1,2-gram)	positive negative	0.9039 (254/281) 0.8487 (202/238)	0.8759 (254/290) 0.8821 (254/290)	0.8897 0.8651	0.8786
SVM (1-gram)	positive negative	0.9055 (249/275) 0.8320 (203/244)	0.8586 (249/290) 0.8865 (203/229)	0.8814 0.8584	0.8709
SVM (1,2-gram)	positive negative	0.9188 (249/271) 0.8347 (207/248)	0.8586 (249/290) 0.9039 (207/229)	0.8877 0.8679	0.8786
$PolaritySim_{Cosine}$	positive negative	0.8571 (222/259) 0.7385 (192/260)	0.7655 (222/290) 0.8384 (192/229)	0.8102 0.7853	0.8087
$PolaritySim_{QACW}, b = 0.5, k_1 = 4$	positive negative	0.9220 (260/282) 0.8734 (207/237)	0.8966 (260/290) 0.9039 (207/229)	0.9091 0.8884	**0.8998**
$PolaritySim_{TF.IDF}$	positive negative	0.9088 (249/ 274) 0.8327 (204/245)	0.8586 (249/290) 0.8908 (204/229)	0.8830 0.8608	0.8728
$PolaritySim_{W2V}$	positive negative	0.9091 (250/275) 0.8361 (204/244)	0.8621 (250/290) 0.8908 (204/229)	0.8850 0.8626	0.8748

[2]We used MNB and SVM implementations in scikit-learn Python library (Pedregosa et al., 2011)

5.4 Restaurant dataset

This is a larger dataset consisting of 606 "nsubj" and "amod" adjective patterns (482 positive and 124 negative), and representing 164 distinct adjectives. It was constructed by randomly extracting 600 reviews from the Restaurant corpus, and labeling positive/negative subjective expressions and their targets in the text. Two annotators were recruited, each labeling a non-overlapping set of 300 reviews. Prior to this, both annotators labeled the same set of 50 reviews with the inter-annotator agreement of 0.82, calculated by using the *agr* metric of (Wiebe et al., 2005). The *agr* metric was used instead of Kappa because the annotators were labelling words and phrases in text, rather than an extracted set of words (as in the AmbAdj dataset), which means that they may disagree on the boundaries of expressions as well as the presence/absence of an annotation. With these types of annotations it is not possible to use Kappa statistic to calculate inter-annotator agreement. Wiebe, Wilson and Cardie (2005) pointed out these annotation problems when there is no pre-defined set of items to label, and suggested the *agr* metric:

$$agr(a||b) = \frac{|A \text{ matching } B|}{|A|} \tag{3}$$

where A is a set of text strings labeled by the annotator *a* as positive or negative, and B are text strings labeled by annotator *b* with the same polarity.

All reviews used in this dataset were removed from the positive and negative reference corpora used to generate $posV_p$ and $negV_p$ vectors. Table 4 summarizes the results.

Table 4: Results on the large Restaurant dataset

Method	Polarity	Precision	Recall	F-measure	Accuracy
MNB (1-gram)	positive negative	0.9289 (444/478) 0.7031 (90/128)	0.9212 (444/482) 0.7258 (90/124)	0.9250 0.7143	0.8812
MNB (1,2-gram)	positive negative	0.9345 (442/473) 0.6992 (93/133)	0.9170 (442/482) 0.7500 (93/124)	0.9257 0.7237	0.8828
SVM (1-gram)	positive negative	0.9515 (432/454) 0.6711 (102/152)	0.8963 (432/482) 0.8226 (102/124)	0.9231 0.7391	0.8812
SVM (1,2-gram)	positive negative	0.9594 (425/443) 0.6503 (106/163)	0.8817 (425/482) 0.8548 (106/124)	0.9189 0.7387	0.8762
$PolaritySim_{Cosine}$	positive negative	0.9405 (316/336) 0.3843 (98/255)	0.6556 (316/482) 0.7903 (98/124)	0.7726 0.5172	0.7005
$PolaritySim_{QACW}, b = 0.4, k_1 = 500$	positive negative	0.9584 (438/457) 0.7388 (99/134)	0.9087 (438/482) 0.7983 (99/124)	0.9329 0.7674	0.9086
$PolaritySim_{TF.IDF}$	positive negative	0.9562 (437/457) 0.7313 (98/134)	0.9066 (437/482) 0.7903 (98/124)	0.9308 0.7597	0.9052
$PolaritySim_{W2V}$	positive negative	0.9626 (438/455) 0.7426 (101/136)	0.9087 (438/482) 0.8145 (101/124)	0.9349 0.7769	**0.9120**

5.5 MP3 and Photography datasets

Two datasets were generated from the Amazon corpus. These datasets contain verbs, adjectives and nouns related with "amod", "nsubj", "nsubjpass", "dobj" and "rcmod" relations to the opinion target word, which can be a noun or pronoun. We selected one product from MP3 and one from Photography categories that have a large number of reviews with high and low ratings. Table 5 lists the products selected.

Table 5: Number of product reviews

Product (corpus)	Number of reviews per rating category				
	1	2	3	4	5
Creative Labs NOMAD MuVo 128 MB MP3 Player (MP3)	49	8	11	25	32
Sharp VLWD255U MiniDV Digital Camcorder (Photo)	48	15	9	36	31

Each review was split into sentences and parsed using Stanford CoreNLP parser. The patterns (evaluation cases) were generated as described in Section 4.1. As a result, 3,329 cases for MP3 and 5,254 for Photography were generated. The cases were presented to the annotators in the same format as given in Table 2. Three annotators were asked to label the cases that have positive or negative valence. Due to the large number of Photography cases, annotators were required to identify positive/negative cases from the top 2000 cases ranked by the review ID. The average agreement (Kappa) between pairs of annotators was 0.79 and 0.85 for MP3 and Photography respectively. The cases upon which at least two of the annotators agreed were then used in the test set, and consist of 592 (339 positive and 253 negative) cases in MP3 and 424 (227 positive and 197 negative) in Photography. The test cases were removed from the positive and negative corpora that were used to generate $posV_p$ and $negV_p$ vectors. The results of the evaluation are presented in Tables 6 and 7.

Table 6: Results on the MP3 dataset

Method	Polarity	Precision	Recall	F-measure	Accuracy
MNB (1-gram)	positive negative	0.8671 (287/331) 0.8008 (209/261)	0.8466 (287/339) 0.8261 (209/253)	0.8567 0.8132	0.8378
MNB (1,2-gram)	positive negative	0.8879 (301/339) 0.8498 (215/253)	0.8879 (301/339) 0.8498 (215/253)	0.8879 0.8498	**0.8716**
SVM (1-gram)	positive negative	0.8847 (284/321) 0.7970 (216/271)	0.8378 (284/339) 0.8538 (216/253)	0.8607 0.8244	0.8446
SVM (1,2-gram)	positive negative	0.8665 (279/322) 0.7778 (210/270)	0.8230 (279/339) 0.8300 (210/253)	0.8442 0.8031	0.8260
$PolaritySim_{Cosine}$	positive negative	0.7370 (213/289) 0.5993 (172/287)	0.6283 (213/339) 0.6798 (172/253)	0.6783 0.6370	0.6684
$PolaritySim_{QACW}, b = 0.1, k_1 = 100$	positive negative	0.8500 (289/340) 0.8347 (197/236)	0.8525 (289/339) 0.7787 (197/253)	0.8513 0.8057	0.8438
$PolaritySim_{TF.IDF}$	positive negative	0.8450 (289/342) 0.8333 (195/234)	0.8525 (289/339) 0.7708 (195/253)	0.8488 0.8008	0.8402
$PolaritySim_{W2V}$	positive negative	0.8529 (290/340) 0.8390 (198/236)	0.8555 (290/339) 0.7826 (198/253)	0.8542 0.8098	0.8472

Table 7: Results on the Photography dataset

Method	Polarity	Precision	Recall	F-measure	Accuracy
MNB (1-gram)	positive negative	0.8114 (185/228) 0.7887 (153/194)	0.8186 (185/226) 0.7806 (153/196)	0.8150 0.7846	0.8009
MNB (1,2-gram)	positive negative	0.8033 (196/244) 0.8315 (148/178)	0.8673 (196/226) 0.7551 (148/196)	0.8340 0.7914	0.8152
SVM (1-gram)	positive negative	0.8326 (184/221) 0.7910 (159/201)	0.8142 (184/226) 0.8112 (159/196)	0.8233 0.8010	0.8128
SVM (1,2-gram)	positive negative	0.8310 (177/213) 0.7656 (160/209)	0.7832 (177/226) 0.8163 (160/196)	0.8064 0.7901	0.7986
$PolaritySim_{Cosine}$	positive negative	0.6971 (145/208) 0.6030 (120/199)	0.6416 (145/226) 0.6122 (120/196)	0.6682 0.6076	0.6511
$PolaritySim_{QACW}, b = 0, k_1 = 500$	positive negative	0.8541 (199/233) 0.8563 (149/174)	0.8805 (199/226) 0.7602 (149/196)	0.8671 0.8054	**0.8550**
$PolaritySim_{TF.IDF}$	positive negative	0.8498 (198/233) 0.8505 (148/174)	0.8761 (198/226) 0.7551 (148/196)	0.8627 0.8000	0.8501
$PolaritySim_{W2V}$	positive negative	0.8498 (198/233) 0.8505 (148/174)	0.8761 (198/226) 0.7551 (148/196)	0.8627 0.8000	0.8501

5.6 SemEval ABSA 2016 dataset

One of the subtasks of SemEval Aspect Based Sentiment Analysis (ABSA) task (Pontiki et al., 2016) is to determine polarity (positive, negative or neutral) for a given opinion target expression (OTE) in a sentence (Restaurant domain in English). The ABSA 2016 test dataset contains 859 OTE/polarity tuples. While this dataset does not let us directly evaluate the accuracy of resolving sentiment polarity of individual words in a sentence, it allows us to see how well the method can be applied to identify polarity

of OTEs based on the polarities of individual words. An OTE can be a word, a multiword expression or NULL (e.g. in the sentence "Well worth it."). First, we follow the same methodology as described in Section 4. Next, we perform two passes through the OTEs in the test set. In the first pass, for each OTE in the test set, we calculate the majority polarity based on all patterns whose target word fully or partially matches the OTE. If no matches have been found, then average polarity is calculated based on all patterns in the current sentence, and if unresolved, on all patterns in the other sentences in the review. In the first pass, the system also records the total number cases in the review that it predicted as positive or negative. In the second pass, if 60% or more cases in the review were predicted as positive in the first pass, the system converts all negative cases into positive, and similarly, if 60% or more cases in the review were predicted as negative, it converts all positive cases into negative. The results are presented in Table 8. The SVM baseline reported in the table is the official task baseline defined by the ABSA organizers, and is described in (Pontiki et al., 2016). Table 8 also lists the results of three top performing systems in the same category (unconstrained) in ABSA 2016. All three systems rely on the ABSA annotated training datasets, and some use other handcrafted resources, such as lexicons (Kumar et al., 2016). We used simple rules for combining individual word polarities to adapt our method to this ABSA task, and did not specifically address neutral polarity and NULL OTE targets. Better techniques to address these issues may lead to further improvements. Even with a simple adaptation to the ABSA task, $PolaritySim_{QACW}$ achieved 83% accuracy, which suggests that an approach using only consumer-rated review corpora is a viable alternative to the existing approaches that require specially constructed resources, such as training datasets and lexicons.

Table 8: Results on the ABSA 2016 test dataset (Restaurant)

Method	Polarity	Precision	Recall	F-measure	Accuracy
ABSA baseline (SVM)	positive negative neutral	0.8132(553/680) 0.5787(103/178) 1(1/1)	0.9051(553/611) 0.5049(103/204) 0.0227(1/44)	0.8567 0.5393 0.0444	0.7648
$PolaritySim_{Cosine}$	positive negative neutral	0.8595(471/548) 0.5276(153/290) 0.0476(1/21)	0.7709(471/611) 0.7500(153/204) 0.0227(1/44)	0.8128 0.6194 0.0308	0.7276
$PolaritySim_{QACW}, b = 0.1, k_1 = 500$	positive negative neutral	0.8734(559/640) 0.7500(153/204) 0.0667(1/15)	0.9149(559/611) 0.7500(153/204) 0.0227(1/44)	0.8937 0.7500 0.0339	**0.8300**
$PolaritySim_{TF.IDF}$	positive negative neutral	0.8262(580/702) 0.7943(112/141) 0.1250(2/16)	0.9493(580/611) 0.5490(112/204) 0.0455(2/44)	0.8835 0.6493 0.0667	0.8079
$PolaritySim_{W2V}$	positive negative neutral	0.8766(554/632) 0.7109(150/211) 0.0625(1/16)	0.9067(554/611) 0.7353(150/204) 0.0227(1/44)	0.8914 0.7229 0.0333	0.8207
IIT-T (Kumar et al., 2016)	–	–	–	–	0.8673
NileT (Khalil and El-Beltagy, 2016)	–	–	–	–	0.8545
IHS-R (Chernyshevich, 2016)	–	–	–	–	0.8394

5.7 Out-of-domain training

While there exist large volumes of consumer-rated reviews in a wide range of domains, some categories of reviews have fewer or even no numerical ratings associated with them, such as reviews written in blogs or forums. Similarly, opinionated content expressed about certain subjects, such as current events or politics, may have no numerical ratings. One way to address the lack of reference corpora in a given domain is to use corpora with user-rated reviews from a different domain. The technique of using a labeled training corpus from one domain to predict labels in a test set from a different domain is known as *out-of-domain training*. To evaluate the performance of our method with out-of-domain reference corpora, we performed a series of experiments, where the positive and negative reference corpora were constructed from a different category of consumer reviews (Movie reviews), while the testing was done on the AmbAdj, Restaurant, MP3 and Photography test datasets described in Sections 5.3−5.5. Specifically, 572,765 documents with 1 and 2 star ratings in the Movie category of Amazon reviews were

extracted to form the negative corpus, and the same number of reviews with 5 star ratings was subsampled to form the positive corpus. They were processed in the same way as the reference corpora described in Section 5.1. The results are presented in Table 9.

System	AmbAdj	Restaurant	MP3	Photo
MNB (1-gram)	0.7360	0.7508	0.6774	0.6161
MNB (1,2-gram)	0.7611	0.7954	0.7500	0.6848
SVM (1-gram)	0.7514	0.7343	0.7365	0.6469
SVM (1,2-gram)	0.7360	0.7657	0.7280	0.6872
$PolaritySim_{QACW}$	**0.8444**	0.8142	0.7856	0.8333
$PolaritySim_{TF.IDF}$	**0.8444**	**0.8209**	**0.7996**	**0.8455**

Table 9: Accuracy of different methods with the out-of-domain training corpus.

The results show that $PolaritySim$ methods outperform all SVM and MNB baselines. Specifically, the accuracy of the best $PolaritySim$ method is higher than the best SVM or MNB run on AmbAdj, Restaurant, MP3 and Photo datasets by 10.9%, 3.2%, 6.6% and 23% respectively. This suggests that the proposed methods are more robust than the state-of-the-art machine learning approaches with the out-of-domain training corpora.

As expected, the use of out-of-domain corpora overall leads to somewhat lower accuracy compared to the use of within-domain training corpora with the same methods (Tables 3−7), although the $PolaritySim$ accuracy is still quite high on every dataset (at or above 80%). The differences between the within-domain and out-of-domain $PolaritySim_{TF.IDF}$ runs on AmbAdj, Restaurant, MP3 and Photo datasets are 3.3%, 10.3%, 5.1% and 0.5% respectively. In contrast, SVM and MNB performance dropped more substantially. For instance, the performance of SVM (1,2-gram) dropped by 19.4%, 14.4%, 13.5% and 16.2%.

6 Results and Discussion

The results in Tables 3-8 show that $PolaritySim_{QACW}$ and $PolaritySim_{TF.IDF}$ are more effective model variants than $PolaritySim_{Cosine}$. Both $PolaritySim_{TF.IDF}$ and $PolaritySim_{QACW}$ outperformed all MNB and SVM baselines on the Restaurant and Photography datasets, as well as the official SVM baseline on the ABSA dataset. On the AmbAdj dataset $PolaritySim_{QACW}$ outperformed the best MNB and SVM variants with unigram and bigram features by 2.4%, whereas $PolaritySim_{TF.IDF}$ showed slightly lower results. On the MP3 dataset, MNB with unigram and bigram features showed the highest results. On the Photography dataset $PolaritySim_{QACW}$ outperformed the best baseline (MNB with unigram and bigram features) by 4.9%. $PolaritySim_{QACW}$ performs better than $PolaritySim_{TF.IDF}$ on all collections, with the most notable differences being 3.1% (AmbAdj) and 2.7% (ABSA). The optimal values for the tuning constant k_1 are mostly very high: 4 (Ambig. adj), 100 (MP3) and 500 (Restaurant, Photo, ABSA), while the optimal *b* values are rather low: 0.5 (Ambig. adj.), 0.1 (MP3, ABSA), 0.4 (Restaurant), and 0 (Photo). In practice, the simpler $PolaritySim_{TF.IDF}$ model may be sufficient for this task, as it does not require tuning constants and has consistently good performance.

As described in Section 3.2, if polarity of a test case is unresolved using $PolaritySim$ due to zero similarity between the vector $EvalV_p$ and both vectors $posV_p$ and $negV_p$, the probability of *p* in positive and negative reference corpora is calculated, and the polarity of the corpus where p has the highest probability of occurrence is assigned. The percentages of test cases with polarity unresolved by $PolaritySim$ method for AmbAdj, Restaurant, MP3, Photography and ABSA datasets are: 0%, 7.95%, 2.4%, 6.88% and 9.9% respectively. Small percentages of unresolved cases show that performance of the system is largely determined by the $PolaritySim$ method. Since the percentages of positive/negative test cases are not the same in each dataset, it is informative to also compare the results in Tables 3-7 to random choice baselines. The random choice baseline accuracy for AmbAdj, Restaurant, Photo and MP3 datasets is 0.5069, 0.6745, 0.5025 and 0.5106 respectively.

Feature word expansion using Word2Vec ($PolaritySim_{W2V}$) led to only small improvements on some datasets. This may be due to the quality of related words, which are often related very broadly. For example, the top five related words for pixelation are artifacting, pixilation, graininess, pixelization, smearing; and for coffee: gumbo, gelato, pastry, donut, malt. While some words are closely related semantically, e.g. pixelation and artifacting, others have a more distant relationship, e.g. coffee and gelato. This can have adverse effect on performance, as a match on a related feature that has weak semantic similarity to the original feature may potentially lead to incorrect polarity attribution. For example, while cold is considered positive when talking about gelato, it is usually negative when referring to coffee.

6.1 Features

The results in Tables 3-8 were obtained using lexico-syntactic features derived from dependency relations at distances ≤ 3 from the given word (Section 4.1.1). These features consist of the lemma, its POS tag and dependency relation (lemma+POS+depRel), extracted from the dependency relation triple of the word in the pattern *p*. It is interesting to see, firstly, whether the use of lexico-syntactic features offers any advantage compared to the use of only lexical features (lemmas) or only syntactic features (POS+depRel), and secondly, whether dependency relations at distances ≤ 3 are useful compared to only direct relations (i.e. distance=1). The accuracy results of the $PolaritySim_{TF.IDF}$ model with different features are presented in Table 10. The use of lexico-syntactic relations (lemma+POS+depRel) as features is better than only lemmas or POS+depRel on all four collections. When only direct relations are used, the performance gains over lemmas on Restaurant, MP3 and Photo datasets are 24.8%, 13.9% and 29.6% respectively. Gains over POS+depRel are 65%, 32.5% and 36.6%. Gains on the ambiguous adjectives dataset are marginal, which may be explained by the small size of this dataset: four distinct adjectives. Having lexico-syntactic relations as features makes intuitive sense. Consider the following sentence: "The price is right, and it has everything that I need." We want to determine polarity of the test case: "NEGATION=0; nsubj; have; VBZ" referring to the target "it". The polarity determined by annotators is positive. One of the features derived from this sentence is "everything; NN; dobj". If we have only lemma features, then we would derive the same feature "everything" from "Everything it has is substandard.", which conveys negative sentiment. If we have only POS+depRel features, then the feature "NN; dobj" would also be derived from "It has hardly anything that I need.", which again has a negative sentiment. If these two sentences were part of the negative corpus, we would have two false matches on these features, potentially causing the assignment of incorrect polarity.

Features	Ambig. Adj.	Restaurant	MP3	Photo
lemma+POS+depRel, $dist = 1$	0.8709	**0.9103**	**0.8403**	0.8428
lemma+POS+depRel, $dist \leq 3$	**0.8728**	0.9052	**0.8403**	**0.8501**
lemma, $dist = 1$	0.8593	0.7293	0.7378	0.6560
lemma, $dist \leq 3$	0.8574	0.7970	0.7587	0.7445
POS+depRel, $dist = 1$	0.8632	0.5516	0.6343	0.6225
POS+depRel, $dist \leq 3$	0.8632	0.7479	0.7123	0.7108

Table 10: Accuracy of $PolaritySim_{TF.IDF}$ method with different features.

Relations at distances ≤ 3 only slightly improve performance compared to direct relations (distance=1) in two out of four datasets with lemma+POS+depRel features. Their removal reduces the feature set size considerably, thus leading to more efficient computation. It is, therefore, more practical to use direct relations only.

6.2 Error analysis

To understand the nature of errors, we performed error analysis based on the results of $PolaritySim_{TF.IDF}$. In the Restaurant domain, 36 positive cases were misclassified as negative. A large proportion of these cases were weakly positive or neutral. For example, nine instances of "decent", three "ok" and one "average" were misclassified as negative. Interestingly, while the annotators labeled

them as positive, they tend to occur more in the negative reviews, and generally convey neutral or weakly positive attitude towards some aspect of an overall negatively reviewed entity, e.g. "The food was decent but I wouldn't go back." Another example labeled as positive, but classified as negative, is "fine" in "The food portions are fine but the plates are so unusually small..." Here, again, one could argue that "fine" is used to express a neutral or weakly positive stance in an overall negative context. The features of the pattern "NEGATION=0; nsubj; fine; JJ" in $negV_p$ that matched the $evalV_p$ vector of this pattern are: "be; VBP; cop" ($TF = 13$) and "small; JJ; conj_but" ($TF = 1$), while the only matching feature in $posV_p$ vector is "be; VBP; cop" ($TF = 5$). Another class of errors is caused by zero matches between $evalV_p$ and both $posV_p$ and $negV_p$. There were 8 such cases out of 36 false negatives, and 13 cases out of 23 false positives in the Restaurant domain.

In some cases a word simply co-occurs more frequently with each of the matching features in the corpus with the polarity that is opposite to the correct polarity of the given instance. An example of such false positive is "cold" in "Teeny tiny slivers of ice cold fish on top of overcooked rice." The matching features of the pattern "NEGATION=0; amod; cold; JJ" in $posV_p$ are: "NN:ice; NN:NN; amod:nn" ($TF = 7$) and "fish; NN; amod" ($TF = 3$), while $negV_p$ only matched "NN:ice; NN:NN; amod:nn" ($TF = 6$). In this example, "ice cold NN" and "cold fish" are more frequent in the positive corpus than negative, which explains why this case was classified as positive.

Another category of errors is caused by the sentence being not specific enough, e.g. "big" in "The camcorder was big." was falsely classified as positive in the Photography dataset. Here, the preceding sentences: "Lowlights: 1. The size. The camcorder is big." contain negative sentiment clues, but the current method does not use them. One possible extension that may alleviate such problems is the addition of patterns from nearby sentences to the feature vector. Another class of errors was due to typos and grammatically incorrect sentences, which caused parser errors and mismatch of features.

6.3 Types of syntactic relations

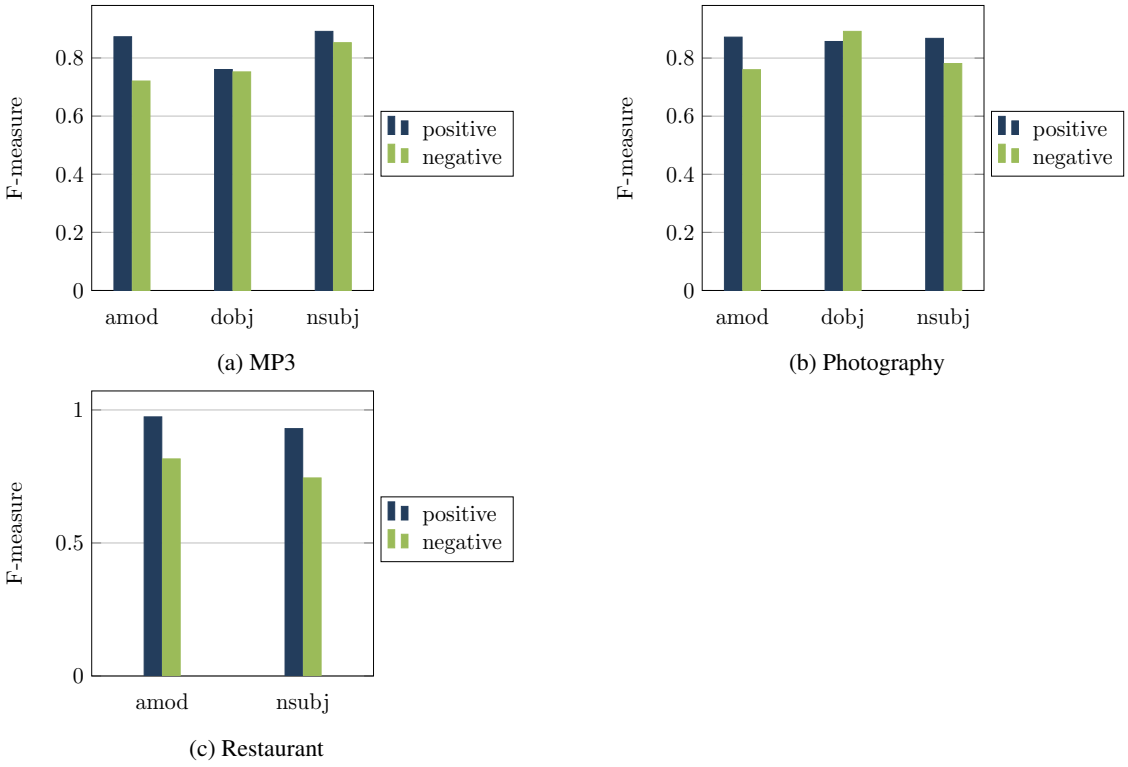
The evaluation datasets are comprised of different syntactic relations between the opinion word and the target word. Table 11 lists the number of evaluation cases broken down by the type of syntactic relation and polarity. It is interesting to note that some syntactic relations occur more frequently in expressions with a certain polarity. For example, nsubj_xcomp is used much more frequently with negative polarity (e.g. "horizontal lines *began to run* through the picture", "Very disappointed that I *have to return* it."). Figure 4 shows how performance (F-measure) varies by the type of syntactic relation between the opinion word and its target for the most frequent syntactic relations. As we can see, there is not much variability in performance, with all categories showing consistently good results. Generally, the performance is higher for positive cases, except for dobj in the Photography domain.

Table 11: Number of evaluation cases by syntactic relationships

Relationship	MP3		Photo		Restaurant	
	pos	neg	pos	neg	pos	neg
amod	123	57	84	53	225	55
dobj	57	62	23	36	0	0
nsubj	131	99	102	80	247	67
nsubj_acomp	17	0	9	3	0	0
nsubjpass	4	8	6	11	2	0
nsubj_xcomp	5	26	0	10	0	0
rcmod	2	1	2	3	8	2

7 Conclusion and future work

We described an effective method called $PolaritySim$ for determining word-level contextual polarity that uses readily available consumer rated reviews as the only external resource. The advantage of $PolaritySim$ is that it does not require manually constructed sentiment lexicons or corpora annotated at word or sentence level, which are labour-intensive resources to build. We approach the problem of



(a) MP3

(b) Photography

(c) Restaurant

Figure 4: Performance by the type of syntactic relation between the opinion word and target

word-level polarity determination as an IR problem, whereby the context vector representing the test case is treated as the "query", while the corresponding word vectors derived from the positive and negative reference corpora are treated as "documents". The method shows improvements compared to the state-of-the-art baselines, SVM and MNB, in three out of four word-level polarity datasets. The method also performs better than the official SVM baseline on the ABSA dataset.

While a large number of consumer-rated reviews is available in a wide range of business and product categories, some categories of reviews and other opinionated content do not have user-assigned polarity labels. We evaluated the performance of $PolaritySim$, MNB and SVM with the out-of-domain reference corpus (Movie reviews) on four datasets: Ambiguous adjectives, Restaurant, MP3 and Photography. An interesting finding is that while the performance of all methods is lower compared to the same methods using within-domain corpora, the performance of $PolaritySim$ dropped much less than that of SVM and MNB. Also, the absolute performance of $PolaritySim$ (at or above 80%) with the out-of-domain corpus is much higher than the performance of SVM and MNB. This suggests that $PolaritySim$ is more robust with the use of out-of-domain corpora than machine learning approaches.

The analysis of different features shows that lexico-syntactic features consisting of lemma+POS+depRel are substantially better than only lemma or POS+depRel. Composite relations at distances ≤ 3 in the dependency graph are not appreciably different compared to direct relations (distance=1) when used with lemma+POS+depRel features. The datasets created as part of this work are available to the research community and can be used for evaluating word-level contextual polarity methods. Some possible future extensions of this work are outlined below.

The approach is currently used to determine binary (positive/negative) polarity. In future work, we will investigate how to adapt it to determine more fine-grained polarity categories.

Contextual polarity of a word may be affected by the factors outside of the current sentence. For instance it may be hard to detect a sarcastic use of the word "great" in the sentence "That's great!" without considering inter-sentential context. The proposed model is extensible. For example, features extracted from other sentences can be added to the vector representation of words.

Acknowledgments

The author would like to thank the following annotators: Mohamad Ahmadi, Kaheer Suleman, Stuart Sullivan, Jack Thomas and Andrew Toulis. This work has been supported by the NSERC Discovery grant.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *in Proc. of LREC*.

Margaret M. Bradley, Peter J. Lang, Margaret M. Bradley, and Peter J. Lang. 1999. Affective norms for english words (anew): Instruction manual and affective ratings.

Caroline Brun, Julien Perez, and Claude Roux. 2016. Xrce at semeval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 277–281, San Diego, California, June. Association for Computational Linguistics.

Caroline Brun. 2012. Learning opinionated patterns for contextual opinion detection. In *24th International Conference on Computational Linguistics*, page 165.

Maryna Chernyshevich. 2016. Ihs-rd-belarus at semeval-2016 task 5: Detecting sentiment polarity using the heatmap of sentence. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 296–300, San Diego, California, June. Association for Computational Linguistics.

Ilia Chetviorkin and Natalia Loukachevitch. 2014. Two-step model for sentiment lexicon extraction from twitter streams. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 67–72, Baltimore, Maryland, June. Association for Computational Linguistics.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240. ACM.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06*, pages 417–422.

Angela Fahrni and Manfred Klenner. 2008. Old wine or warm beer: Target-specific sentiment analysis of adjectives. In *Proc. of the Symposium on Affective Language in Human and Machine, AISB*, pages 60–63.

Vasileios Hatzivassiloglou and Kathleen R McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics*, pages 174–181. Association for Computational Linguistics.

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*, pages 168–177. ACM.

Daisuke Ikeda, Hiroya Takamura, Lev-Arie Ratinov, and Manabu Okumura. 2008. Learning to shift the polarity of words for sentiment classification.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics.

Mahesh Joshi and Carolyn Penstein-Rose. 2009. Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 313–316. Association for Computational Linguistics.

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

Hinrich Kessler and Wiltrud Schuetze. 2012. Classification of inconsistent sentiment words using syntactic constructions. In *24th International Conference on Computational Linguistics*, pages 569–578.

Talaat Khalil and Samhaa R. El-Beltagy. 2016. Niletmrg at semeval-2016 task 5: Deep convolutional neural networks for aspect category and sentiment extraction. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 271–276, San Diego, California, June. Association for Computational Linguistics.

Ayush Kumar, Sarah Kohail, Amit Kumar, Asif Ekbal, and Chris Biemann. 2016. Iit-tuda at semeval-2016 task 5: Beyond sentiment lexicon: Combining domain dependency and distributional semantics features for aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1129–1135, San Diego, California, June. Association for Computational Linguistics.

Raymond Yiu Keung Lau, Chun Lam Lai, Peter B. Bruza, and Kam F. Wong. 2011. Leveraging web 2.0 data for scalable semi-supervised learning of domain-specific sentiment lexicons. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2457–2460, New York, NY, USA. ACM.

Raymond Y. K. Lau, Stephen S. Y. Liao, K. F. Wong, and Dickson K. W. Chiu. 2012. Web 2.0 environmental scanning and adaptive decision support for business mergers and acquisitions. *MIS Q.*, 36(4):1239–1268, December.

Shoushan Li, Zhongqing Wang, Sophia Yat Mei Lee, and Chu-Ren Huang. 2013. Sentiment classification with polarity shifting detection. In *Asian Language Processing (IALP), 2013 International Conference on*, pages 129–132. IEEE.

Bing Liu. 2015. *Sentiment Analysis - Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *Proceedings of the 20th international conference on World wide web*, pages 347–356. ACM.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Saif M Mohammad. 2015. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. *Emotion Measurement*.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using crfs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 786–794. Association for Computational Linguistics.

F. A. Nielsen. 2011. Afinn. *Technical report. Informatics and Mathematical Modelling, Technical University of Denmark*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124, Stroudsburg, PA, USA. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Livia Polanyi and Annie Zaenen, 2006. *Contextual Valence Shifters*, pages 1–10. Springer Netherlands, Dordrecht.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado*, pages 486–495.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Núria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California, June. Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2015. On the automatic learning of sentiment lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1397–1402, Denver, Colorado, May–June. Association for Computational Linguistics.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Karen Sparck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information Processing & Management*, 36(6):809–840.

Kaheer Suleman and Olga Vechtomova. 2016. Discovering aspects of online consumer reviews. *Journal of Information Science*, 42(4):492–506.

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building large-scale twitter-specific sentiment lexicon : A representation learning approach. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 172–182, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

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.

Olga Vechtomova and Stephen E Robertson. 2012. A domain-independent approach to finding related entities. *Information Processing & Management*, 48(4):654–670.

Olga Vechtomova, Kaheer Suleman, and Jack Thomas. 2014. An information retrieval-based approach to determining contextual opinion polarity of words. In *Proceedings of the 36th European Conference on Information Retrieval (ECIR)*, pages 553–559. Springer.

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior Research Methods*, 45(4):1191–1207.

Albert Weichselbraun, Stefan Gindl, and Arno Scharl. 2013. Extracting and grounding context-aware sentiment lexicons. *IEEE Intelligent Systems*, 28(2):39–46.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

Yunfang Wu and Miaomiao Wen. 2010. Disambiguating dynamic sentiment ambiguous adjectives. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1191–1199. Association for Computational Linguistics.

Rui Xia, Feng Xu, Jianfei Yu, Yong Qi, and Erik Cambria. 2016. Polarity shift detection, elimination and ensemble. *Inf. Process. Manage.*, 52(1):36–45, January.

Zhihua Zhang and Man Lan. 2015. Ecnu: Extracting effective features from multiple sequential sentences for target-dependent sentiment analysis in reviews. *SemEval-2015*, page 736.