SentiCircles for Contextual and Conceptual Semantic Sentiment Analysis of Twitter

Hassan Saif,[1] Miriam Fernandez,[1] Yulan He[2] and Harith Alani[1]

[1] Knowledge Media Institute, The Open University, United Kingdom
{h.saif, m.fernandez, h.alani}@open.ac.uk
[2] School of Engineering and Applied Science, Aston University, UK
y.he@cantab.net

Abstract. Lexicon-based approaches to Twitter sentiment analysis are gaining much popularity due to their simplicity, domain independence, and relatively good performance. These approaches rely on sentiment lexicons, where a collection of words are marked with fixed sentiment polarities. However, words' sentiment orientation (positive, neural, negative) and/or sentiment strengths could change depending on context and targeted entities. In this paper we present SentiCircle; a novel lexicon-based approach that takes into account the contextual and conceptual semantics of words when calculating their sentiment orientation and strength in Twitter. We evaluate our approach on three Twitter datasets using three different sentiment lexicons. Results show that our approach significantly outperforms two lexicon baselines. Results are competitive but inconclusive when comparing to state-of-art SentiStrength, and vary from one dataset to another. SentiCircle outperforms SentiStrength in accuracy on average, but falls marginally behind in F-measure.

Keywords: #eswc2014Saif, Sentiment analysis, Semantics, Twitter

1 Introduction

With over 500 million users and 400 million tweets daily, Twitter has now become a goldmine for monitoring the sentiment of the crowd. Most current approaches for identifying the sentiment of tweets can be categorised into one of two main groups: *supervised approaches* [15, 4, 12], which use a wide range of features and labelled data for training sentiment classifiers, and *lexicon-based approaches* [22, 14, 6], which make use of pre-built lexicons of words weighted with their sentiment orientations to determine the overall sentiment of a given text. Some of these methods tend to achieve good and consistent level of accuracy when applied to well known domains and datasets, where labelled data is available for training, or when the analysed text is well covered by the used sentiment lexicon.

Popularity of lexicon-based approaches is rapidly increasing since they require no training data, and hence are more suited to a wider range of domains than supervised approaches [22]. Nevertheless, lexicon-based approaches have two main limitations. Firstly, the number of words in the lexicons is finite, which may constitute a problem when extracting sentiment from very dynamic environments such as Twitter, where new terms, abbreviations and malformed words constantly emerge. Secondly and more importantly, sentiment lexicons tend to assign a fixed sentiment orientation and score to words, irrespective of how these words are used in the text. Words could express a different sentiment in different contexts. For example, the word “great” should be negative in the context of a “problem”, and positive in the context of a “smile”.

In this paper we propose an approach called SentiCircles, which builds a dynamic representation of context to tune the pre-assigned strength and polarity of words in the lexicon. This approach incorporates two types of semantics; *contextual semantics*, i.e., semantics inferred from the co-occurrence of words [27], and *conceptual semantics*, i.e., semantics extracted from background ontologies such as DBpedia.

Contextual semantics (aka statistical semantics) [27] has been traditionally used in diverse areas of computer science including Natural Language Processing and Information Retrieval [25]. The main principle behind the notion of contextual semantics comes from the dictum-“You shall know a word by the company it keeps!” [10]. This suggests that words that co-occur in a given context tend to have certain relation or semantic influence, which we try to capture with our SentiCircle approach.

We evaluate our approach using three different sentiment lexicons and with three different datasets, and compare its performance against various lexicon baseline methods. Our results show that our SentiCircle approach outperforms the other lexicon methods by nearly 20% in accuracy and 30-40% in F-measure. We also compare our approach against SentiStrength [22], which, to our knowledge, is the leading lexicon-based sentiment detection approach for social media. Our approach outperformed SentiStrength in accuracy in 2 datasets, and in F-measure in one dataset only (detailed later).

The main contributions of this paper can be summarised as follows:

- Introduce a novel lexicon-based approach using a contextual representation of words, called SentiCircles, which is able to capture the implicit semantics of words from their concurrence [27], and to update their sentiment orientation accordingly.
- Conduct several experiments and test the effectiveness of our proposed approach for sentiment detection of tweets against several state-of-the-art baselines.
- Propose two different methods of employing SentiCircles for sentiment detection of tweets and evaluate their effectiveness against other baselines.
- Incorporate conceptual semantics into SentiCircle and study their impact on sentiment detection performance.

In the rest of this paper, related work is discussed in Section 2, and SentiCircle approach and deployment is presented in Sections 3 and 4. Experiments and results are presented in Sections 5 and 6 respectively. Discussion and future work are covered in Section 7. Finally, we conclude our work in Section 8.

2 Related Work

Most existing approaches to Twitter sentiment analysis focus on classifying the individual tweets into subjective (positive or negative) or objective (neutral). They can be categorised as *supervised approaches* and *lexicon-based approaches*.

Supervised approaches are based on training classifiers from various combinations of features such as word n-grams [15, 5], Part-Of-Speech (POS) tags [4, 1], and tweets syntax features (e.g., hashtags, retweets, punctuations, etc.) [12]. These methods can achieve 80%-84% in accuracy [17]. However, training data is usually expensive to obtain [13] especially for continuously evolving subject domains as in Twitter. Furthermore, classifiers trained on data on a specific domain (e.g., movie reviews) may produce low performance when applied to a different domain (e.g., camera reviews) [2].

Lexicon-based methods use the sentiment orientation of opinionated words (e.g., great, sad, excellent) found in a given text to calculate its overall sentiment [14, 6]. Instead of using training data to learn sentiment, lexicon-based methods rely on pre-built dictionaries of words with associated sentiment orientations [20], such as SentiWordNet [3] or the MPQA subjectivity lexicon [26]. Thelwall et al. [23, 22] proposed SentiStrength; a lexicon-based method for sentiment detection on the social web. This method overcomes the common problem of ill-formed language on Twitter and the like, by applying several lexical rules, such as the existence of emoticons, intensifiers, negation and booster words (e.g., absolutely, extremely).

Lexicon-based methods not only provide sentiment polarity (positive/negative), but also *strength*. For example, SentiStrength computes the positive sentiment strength in the range from 1 (not positive) to 5 (extremely positive). One limitation of lexicons is their static sentiment values of terms, regardless of their contexts. Although authors in [23] proposed an algorithm to update the sentiment strength assigned to terms in a lexicon, this algorithm required training from manually annotated corpora.

Another common problem with the above approaches is their full dependence on the presence of words or syntactical features that explicitly reflect sentiment. In many cases however, the sentiment of a word is implicitly associated with the semantics of its context [7]. Several methods have been proposed for exploring semantics for sentiment analysis, which can be categorised into *contextual semantic approaches*, and *conceptual semantic approaches*.

Contextual semantic approaches determined semantics from the co-occurrence patterns of words, which is also known as *statistical semantics* [25, 27], and have often been used for sentiment analysis [24, 21].

Conceptual semantic approaches use external semantic knowledge bases (e.g., ontologies and semantic networks) with NLP techniques to capture the conceptual representations of words that implicitly convey sentiment. In our previous work we showed that incorporating general conceptual semantics (e.g., “president”, “company”) into supervised classifiers improved sentiment accuracy [18]. SenticNet [8],[3] is a concept-based lexicon for sentiment analysis. It contains 14k fine-grained concepts collected from the Open Mind corpus and coupled with their sentiment orientations. SenticNet was proved valuable for sentiment detection in conventional text (e.g., product reviews) [11]. Unlike SentiStrength [23], SenticNet is not tailored for Twitter and the like. Although conceptual semantic approaches have been shown to outperform purely syntactical approaches [7], they are usually limited by the scope of their underlying knowledge bases, which is especially problematic when processing general Twitter streams, with their rapid semiotic evolution and language deformations.

To address the limitations above, we developed SentiCircle, which (1) is based on a lexicon and hence can be applied to data of different domains, (2) captures the contextual semantics of words to update their sentiment orientation and strength, and (3) allows for conceptual semantics to be added to enrich the sentiment analysis task.

[3] http://sentic.net/

3 Capturing and Representing Semantics for Sentiment Analysis

In the following we explain the SentiCircle approach and its use of contextual and conceptual semantics. The main idea behind our SentiCircle approach is that the sentiment of a term is not static, as in traditional lexicon-based approaches, but rather depends on the context in which the term is used, i.e., it depends on its contextual semantics. We define context as a textual corpus or a set of tweets.

To capture the contextual semantics of a term we consider its co-occurrence patterns with other terms, as inspired by [27]. Following this principle, we compute the semantics of a term m by considering the relations of m with all its context words (i.e., words that occur with m in the same context). To compute the individual relation between the term m and a context term c_i we propose the use of the *Term Degree of Correlation (TDOC)* metric. Inspired by the TF-IDF weighting scheme this metric is computed as:

$$TDOC(m, c_i) = f(c_i, m) \times \log \frac{N}{N_{c_i}} \qquad (1)$$

where $f(c_i, m)$ is the number of times c_i occurs with m in tweets, N is the total number of terms, and N_{c_i} is the total number of terms that occur with c_i. In addition to each TDOC computed between m and each context term c_i, we also consider the *Prior Sentiment* of c_i, extracted from a sentiment lexicon. As with common practice, if this term c_i appears in the vicinity of a negation, its prior sentiment score is negated. The negation words are collected from the General Inquirer under the NOTLW category.[4]

3.1 Representing Semantics with SentiCircles

Contextual semantics of a term m are represented as a geometric circle; *SentiCircle*, where the term is situated in the centre of the circle, and each point around it represents a context term c_i. The position of c_i is defined jointly by its prior sentiment and its degree of correlation (TDOC). The rational behind using this circular representation shape, which will become clearer later, is to benefit from the trigonometric properties it offers for estimating the sentiment orientation, and strength, of terms. It also enables us to calculate the impact of context words on the sentiment orientation and on the sentiment strength of a target-word separately, which is difficult to do with traditional vector representations. Formally, a SentiCircle in a polar coordinate system can be represented with the following equation:

$$r^2 - 2rr_0cos(\theta - \phi) + r_0^2 = a^2 \qquad (2)$$

where a is the radius of the circle, (r_0, ϕ) is the polar coordinate of the centre of the circle, and (r, θ) is the polar coordinate of a co-occurring term on the circle. For simplicity, we assume that our SentiCircles are centred at the origin (i.e., $r_0 = 0$).

Hence, to build a SentiCircle for a term m, we only need to calculate, for each context term c_i a radius r_i and an angle θ_i. To do that, we use the prior sentiment score and the TDOC value of the term c_i as:

$$\begin{aligned} r_i &= \text{TDOC}(m, c_i) \qquad (3) \\ \theta_i &= \text{Prior\_Sentiment}(c_i) * \pi \end{aligned}$$

[4] http://www.wjh.harvard.edu/~inquirer/NotLw.html

We normalise the radii of all terms in a SentiCircle to a scale between 0 and 1. Hence, the radius a of any SentiCircle is equal to 1. Also, all angles' values are in radian.

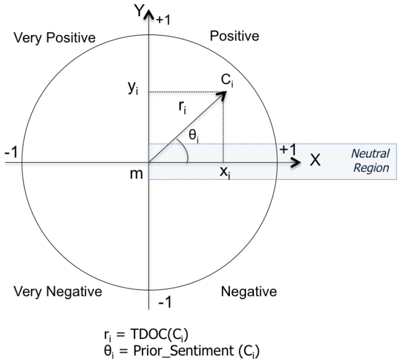


Fig. 1. SentiCircle of a term m

The SentiCircle in the *polar coordinate system* can be divided into four sentiment quadrants as shown in Figure 1. Terms in the two upper quadrants have a positive sentiment ($\sin\theta > 0$), with upper left quadrant representing stronger positive sentiment since it has larger angle values than those in the top right quadrant. Similarly, terms in the two lower quadrants have negative sentiment values ($\sin\theta < 0$). Although the radius of the SentiCircle of any term m equals to 1, points representing context terms of m in the circle have different radii ($0 \leq r_i \leq 1$), which reflect how important a context term is to m. The larger the radius, the more important the context term to m.

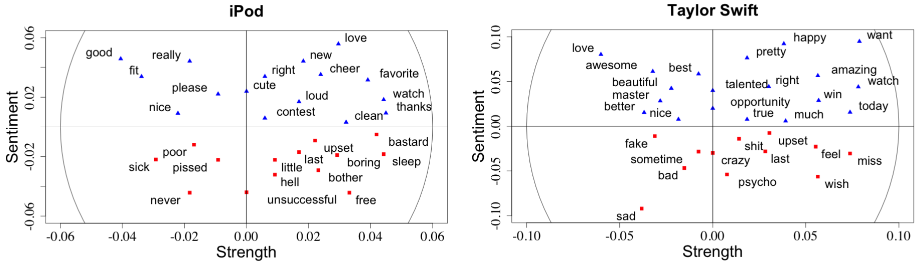
We can move from the *polar coordinate system* to the *Cartesian coordinate system* by simply using the trigonometric functions `sine` and `cosine` as:

$$x_i = r_i \cos\theta_i \qquad\qquad y_i = r_i \sin\theta_i \tag{4}$$

Moving to the *Cartesian coordinate system* allows us to use the trigonometric properties of the circle to encode the contextual semantics of a term in the circle as sentiment orientation and sentiment strength. Y-axis in the Cartesian coordinate system defines the sentiment of the term, i.e., a positive y value denotes a positive sentiment and vice versa. The X-axis defines the sentiment strength of the term. The smaller the x value, the stronger the sentiment.[5] Moreover, a small region called the "*Neutral Region*" can be defined. This region, as shown in Figure 1, is located very close to X-axis in the "*Positive*" and the "*Negative*" quadrants only, where terms lie in this region have very weak sentiment (i.e., $|\theta| \approx 0$). The "*Neutral Region*" has a crucial role in measuring the overall sentiment of a given SentiCircle as will be shown in the subsequent sections.

Note that in the extreme case, where $r_i = 1$ and $\theta_i = \pi$ we position the context term c_i in the "*Very Positive*" or the "*Very Negative*" quadrants based on the sign of its prior sentiment score.

[5] This is because $\cos\theta < 0$ for large angles.



(a) iPod.

(b) Taylor Swift.

Fig. 2. Example SentiCircles for "`iPod`" and "`Taylor Swift`". We removed points near the origin to ease visualisation. Dots in the upper half of the circle (triangles) represent terms bearing a positive sentiment while dots in the lower half (squares) are terms with a negative sentiment.

Figure 2 shows the SentiCircles of the entities "`iPod`" and "`Taylor Swift`". Terms (i.e., points) inside each circle are positioned in a way that represents their sentiment scores and their importance (degree of correlation) to the entity. For example, "`Awesome`" in the SentiCircle of "`Taylor Swift`" has a positive sentiment and a high importance score, hence it is positioned in the "*Very Positive*" quadrant (See Figure 2(b)). The word "`Pretty`", in the same circle, also has positive sentiment, but it has lower importance score than the word "`Awesome`", hence it is positioned in the "*Positive*" quadrant. We also notice that there are some words that appear in both circles, but in different positions. For example, the word "`Love`" has a stronger positive sentiment strength with "`Taylor Swift`" compared to "`iPod`", although it has a positive sentiment (similar y-value) in both circles.

As described earlier, the contribution of both quantities (prior sentiment and degree of correlation) is calculated and represented in the SentiCircle separately by means of the projection of the context term along X-axe (sentiment strength) and Y-axe (sentiment orientation). Such level of granularity is crucial when we need, for example, to filter those context words that have low contribution towards the sentiment orientations or strength of the target word.

3.2 Using SentiCircles to Measure Sentiment

The above examples show that, although we use external lexicons to assign initial sentiment scores to terms, our SentiCircle representation is able to amend these scores according to the context in which each term is used. To compute the new sentiment of the term based on its SentiCircle we use the *Senti-Median* metric. We now have the SentiCircle of a term m which is composed by the set of (x, y) Cartesian coordinates of all the context terms of m, where the y value represents the sentiment and the x value represents the sentiment strength. An effective way to approximate the overall sentiment of a given SentiCircle is by calculating the geometric median of all its points. Formally, for a given set of n points $(p_1, p_2, ..., p_n)$ in a SentiCircle Ω, the 2D geometric median g is defined as:

$$g = \arg\min_{g \in \mathbb{R}^2} \sum_{i=1}^{n} |||p_i - g||_2, \qquad (5)$$

where the geometric median is a point $g = (x_k, y_k)$ in which its Euclidean distances to all the points p_i is minimum. We call the geometric median g the Senti-Median as it captures the sentiment (y-coordinate) and the sentiment strength (x-coordinate) of the SentiCircle of a given term m.

Following the representation provided in Figure 1, the sentiment of the term m is dependent on whether the Senti-Median g lies inside the neutral region, the positive quadrants, or the negative quadrants. Formally, given a Senti-Median g_m of a term m, the term-sentiment function $\mathcal{L}$ works as:

$$\mathcal{L}(g_m) = \begin{cases} \text{negative} & \text{if } y_g < -\lambda \\ \text{positive} & \text{if } y_g > +\lambda \\ \text{neutral} & \text{if } |y_g| \leq \lambda \;\&\; x_g \leq 0 \end{cases} \tag{6}$$

where λ is the threshold that defines the Y-axis boundary of the neutral region. Section 5 illustrates how this threshold is computed.

3.3 Enriching SentiCircles with Conceptual Semantics

We take conceptual semantics to refer to the semantic concepts (e.g., "`person`", "`company`", "`city`") that represent entities (e.g., "`Steve Jobs`", "`Vodafone`", "`London`") appearing in tweets. In this section we describe the addition of conceptual semantics into the SentiCircle representation.

As in our previous work [18], AlchemyAPI[6] came first amongst the set of entity extractors we tested on Twitter. Here we use AlchemyAPI again to extract all named entities in tweets with their associated concepts. We add the concepts into the SentiCircle representation using the **Semantic Augmentation** method [18], where we add the semantic concepts to the original tweet before applying our representation model (e.g., "`headache`" and its concept "`Health Condition`" will appear together in the SentiCircle). Also note that each extracted concept will be represented by a SentiCircle in order to compute its overall sentiment.

The rational behind adding these concepts is that certain entities and concepts tend to have a more consistent correlation to terms of positive or negative sentiment. This can help determining the sentiment of semantically relevant or similar entities which do not explicitly express sentiment. In the example in Figure 4, "`Wind`" and "`Humidity`" have negative SentiCircles as they tend to appear with negative terms in tweets. Hence their concept "`Weather Condition`" will have a negative sentiment. The tweet "`Cycling under a heavy rain.. What a #luck!`" is likely to have a negative sentiment due to the presence of the word "`rain`" which is mapped to the negative concept "`Weather Condition`". Moreover, the word `heavy` in this context is more likely to have a negative sentiment due to its correlation with "`rain`" and "`Weather Condition`".

4 Using SentiCircles for Tweet-level Sentiment Analysis

There are several ways in which the SentiCircle representations of the terms in a tweet can be used to determine the tweet's overall sentiment. For example, the tweet "`iPhone and iPad are amazing`" contains five terms. Each of these terms has an associated

[6] `www.alchemyapi.com`

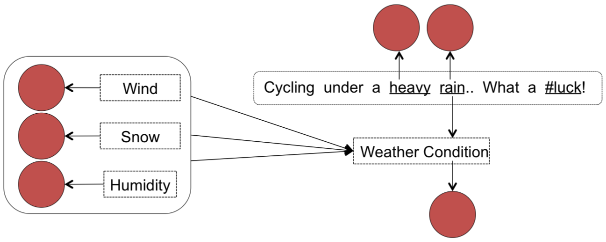


Fig. 3. Mapping semantic concepts to detect sentiment.

SentiCircle representation, which can be combined in different ways to extract the tweet's sentiment. We experiment with two ways for using SentiCircle representations for tweet-level sentiment detection:

Median Method: This method takes the median of all Senti-Medians, and this assumes all tweet terms to be equal. Each tweet $t_i \in \mathcal{T}$ is turned into a vector of Senti-Medians $\boldsymbol{g} = (g_1, g_2, ..., g_n)$ of size n, where n is the number of terms that compose the tweet and g_j is the Senti-Median of the SentiCircle associated with term m_j. Equation 5 is used to calculate the median point q of $\boldsymbol{g}$, which we use to determine the overall sentiment of tweet t_i using Function 6.

Pivot Method: This method favours some terms in a tweet over others, based on the assumption that sentiment is often expressed towards one or more specific targets, which we refer to as "`Pivot`" terms. In the tweet example above, there are two pivot terms, "`iPhone`" and "`iPad`" since the sentiment word "`amazing`" is used to describe both of them. Hence, the method works by (1) extracting all pivot terms in a tweet and; (2) accumulating, for each sentiment label, the sentiment impact that each pivot term receives from other terms. The overall sentiment of a tweet corresponds to the sentiment label with the highest sentiment impact. Opinion target identification is a challenging task and is beyond the scope of our current study. For simplicity, we assume that the pivot terms are those having the POS tags: {*Common Noun, Proper Noun, Pronoun*} in a tweet. For each candidate pivot term, we build a SentiCircle from which the sentiment impact that a pivot term receives from all the other terms in a tweet can be computed. Formally, the Pivot-Method seeks to find the sentiment $\hat{s}$ that receives the maximum sentiment impact within a tweet as:

$$\hat{s} = \arg\max_{s \in \mathcal{S}} \mathcal{H}_s(\boldsymbol{p}) = \arg\max_{s \in \mathcal{S}} \sum_{i}^{N_{\boldsymbol{p}}} \sum_{j}^{N_{\boldsymbol{w}}} \mathcal{H}_s(p_i, w_j) \tag{7}$$

where $s \in \mathcal{S} = \{Positive, Negative, Neutral\}$ is the sentiment label, $\boldsymbol{p}$ is a vector of all pivot terms in a tweet, $N_{\boldsymbol{p}}$ and $N_{\boldsymbol{w}}$ are the sets of the pivot terms and the remaining terms in a tweet respectively. $\mathcal{H}_s(p_i, w_j)$ is the sentiment impact function, which returns the sentiment impact of a term w_j in the SentiCircle of a pivot term p_i. The sentiment impact of a term within a SentiCircle of a pivot term is the term's Euclidean distance from the origin (i.e., the term's radius). Note that the impact value is doubled for all terms located either in the "*Very Positive*" or in the "*Very Negative*" quadrants.

If the Pivot method fails to detect a pivot term (e.g., if tweet is too short or has many ill-formed words), or finds a zero sentiment impact for all pivot terms (e.g., $N_{\boldsymbol{w}}$ terms are positioned at the origin (0,0)), then the method will revert back to the Median method.

5 Experimental Setup

As mentioned in Section 4 the contextual semantics captured by the SentiCircle representation are based on terms co-occurrence from the corpus and an initial set of sentiment weights from a sentiment lexicon. We propose an evaluation set up that uses three different corpora (collections of tweets) and three different generic sentiment lexicons.

Datasets: We use three Twitter datasets which have been used in other sentiment analysis literature. Numbers of positive and negative tweets within these datasets are summarised in Table 1, and detailed in the references added in the table.

Dataset	Tweets	Positive	Negative
Obama McCain Debate (OMD)[9]	1081	393	688
Health Care Reform (HCR)[19]	1354	397	957
Standford Sentiment Gold Standard (STS-Gold)[16]	2034	632	1402

Table 1. Twitter datasets used for the evaluation

Sentiment Lexicons: As describe in Section 4, initial sentiments of terms in SentiCircle are extracted from a sentiment lexicon (prior sentiment). We evaluate our approach using three external sentiment lexicons in order to study how the different prior sentiment scores of terms influence the performance of the SentiCircle representation for sentiment analysis. The aim is to investigate the ability of SentiCircles in updating these *context-free* prior sentiment scores based on the contextual semantics extracted from different tweets corpora. We selected three state-of-art lexicons for this study: (i) the SentiWordNet lexicon [3], (ii) the MPQA subjectivity lexicon [26] and, (iii) Thelwall-Lexicon [23, 22].

Baselines: We compare the performance of SentiCircle in sentiment polarity detection of tweets (positive vs. negative) against the following baselines:

- *Lexicon labelling:* Use the MPQA (hereafter *MPQA-Method*) and the SentiWordNet (hereafter *SentiWordNet-Method*) lexicons to extract sentiment. If a tweet contains more positive words than negative ones, it is labelled as positive, and vice versa.
- *SentiStrength* [23, 22]: is a state-of-the-art approach, which assigns to each tweet two sentiment strengths: a negative strength between -1 (not negative) to -5 (extremely negative) and a positive strength between +1 (not positive) to +5 (extremely positive). A tweet is considered positive if its positive sentiment strength is 1.5 times higher than the negative one, and negative otherwise.[7] Note that SentiStrength come with manually-defined lexical rules, such as the existence of emoticons, intensifiers, negation and booster words (e.g., absolutely, extremely), to compute the average sentiment strength of a tweet.

Thresholds and Parameters Tuning: When computing sentiment in a SentiCircle (Function 6) it is necessary to set the geometric boundaries of the neutral region where neutral terms reside. While the boundaries of the neutral region are fixed for the X-axis $[0, 1]$ (see Section 4), the boundaries of the Y-axis need to be determined. Neutral areas

[7] `http://sentistrength.wlv.ac.uk/documentation/SentiStrengthJavaManual.doc`

tend to have a high density of terms, since the number of neutral terms is usually larger than the number of positive and negative terms.

The limits of the neutral region vary from one SentiCircle to another. For simplicity, we assume the same neutral region boundary for all SentiCircles emerging from the same corpus and sentiment lexicon. To compute these thresholds we first build the SentiCircle of the complete corpus by merging all SentiCircles of each individual term and then we plot the density distribution of the terms within the constructed SentiCircle. The boundaries of the neutral are delimited by an increase/decrease in the density of terms.

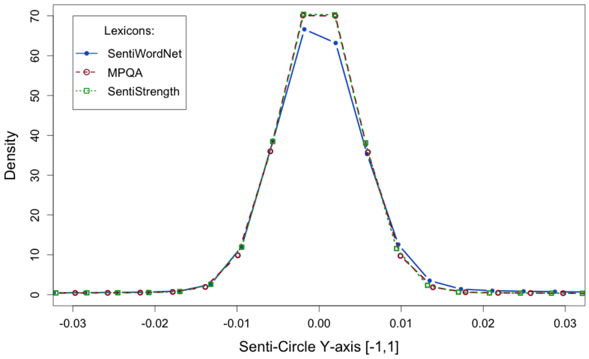


Fig. 4. Density geometric distribution of terms on the OMD dataset.

	SentiWordNet	MPQA	Thelwall-Lexicon
OMD	[-0.01, 0.01]	[-0.01, 0.01]	[-0.01, 0.01]
HCR	[-0.1, 0.1]	[-0.05, 0.05]	[-0.05, 0.05]
STS-Gold	[-0.1, 0.1]	[-0.05, 0.05]	[-0.001, 0.001]

Table 2. Neutral region boundaries for Y-axis.

Figure 4 shows the three density distribution plots for the OMD dataset with SentiWordNet, MPQA and Thelwall lexicons. The boundaries of the neutral area are delimited by the density increase, falling in the $[-0.01, 0.01]$ range. Note that the generated SentiCircles vary depending on corpus and sentiment lexicon. For evaluation, we computed nine neutral regions, one for each corpus and sentiment lexicon used (see Table 2).

6 Evaluation Results

In this section we report the results from using SentiCircle to identify tweet sentiment, with all three methods described in Section 4, using SentiWordNet, MPQA and Thelwall lexicons on OMD, HCR and STS-Gold datasets. We compare our results with those obtained from the baselines described in Section 5. Later we report on the impact of adding conceptual semantics to the analysis (Section 3.3).

We report these results in two different settings. In the first setting, only contextual semantics are considered when constructing the SentiCircle representation. In our second setting, conceptual semantics are added to the SentiCircle representation. Our aim is to study up to which level the introduction of more fine-grained conceptual semantics can help to enrich the contextual semantics for sentiment analysis.

6.1 Results of Sentiment Detection with Contextual Semantics

Figure 5 shows the results in accuracy (left column) and average F-measure (F1-score) (right column) of all the methods and across all three datasets. The significantly worst performing baselines are the ones based solely on lexicons: the MPQA and SentiWordNet lexicons. Remember that SentiStrength adds a wide range of rules on top of the lexicon.

SentiCircle consistently achieved better results when using the MPQA or Thelwall lexicons than SentiWordNet. We also notice generally better results of SentiCircle when favouring target terms in tweets (Pivot method - Section 4), demonstrating good potential of such an approach.

The results show a close competition between our SentiCircle method and the SentiStrength method. For the OMD dataset, SentiCircle outperforms SentiStrength by 5.6% in accuracy (70.58 / 66.79 = 1.056) and 9% in F-measure (66.94 / 61.4 = 1.09) when using MPQA lexicon. For HCR, SentiCircle achieves 5.5% higher accuracy, whereas SentiStrength provides a 1.2% better F-measure. As for STS-Gold, SentiStrength gives around a 1.2% win in both accuracy and F-measure. The average accuracy of SentiCircle and SentiStrength across all three datasets is 72.39 and 71.7 respectively, and for F-measure it is 65.98 and 66.52. Also, the average precision and recall for SentiCircle are %66.82 and %66.12 and for SentiStrength are %67.07 %66.56 respectively.

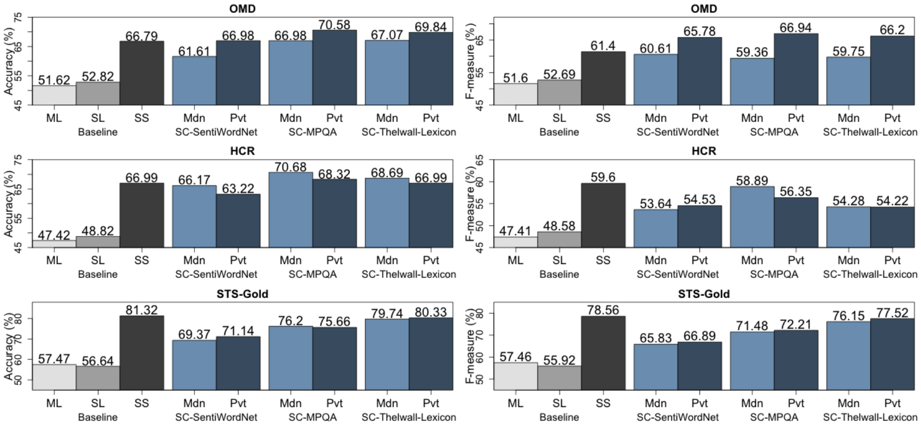


Fig. 5. Sentiment detection results (Accuracy and F-measure). ML: MPQA lexicon, SL: SentiWordNet lexicon, SS: SentiStrength, SC is SentiCircle approach, SC-SentiWordNet: SC with SentiWordNet lexicon, Mdn: SentiCircle with Median method, Pvt: SentiCircle with Pivot method.

Although the potential is evident, clearly there is a need for more research to determine the specific conditions under which SentiCircle performs better or worse. One likely factor that influences the performance of SentiCircle is the balance of positive to negative tweets in the dataset. For example, we notice that SentiCircle produces, on average, 2.5% lower recall than SentiStrength on positive tweet detection. This is perhaps not surprising since our tweet data contain more negative tweets than positive ones with the number of the former more than double the number of the latter (see Table 1).

Remember that the motivation behind SentiCircle is that sentiment of words may vary with context. By capturing the contextual semantics of these words, using the SentiCircle representation, we aim to adapt the strength and polarity of words. We show here the average percentage of words in our three datasets for which SentiCircle changed their prior sentiment orientation or strength.

Table 3 shows that on average 27.1% of the unique words in our datasets were covered by the sentiment lexicons and were assigned prior sentiments accordingly. Using the SentiCircle representation, however, resulted in 59.9% of these words flipping their sentiment orientations (e.g., from positive to negative, or to neutral) and 37.43% changing their sentiment strength while keeping their prior sentiment orientation. Hence only 2.67% of the words were left with their prior sentiment orientation and strength unchanged. It is also worth noting that our model was able to assign sentiment to 38.93% of the *hidden* words that were not covered by the sentiment lexicons. In future work we plan to investigate these results further to understand the influence of these type of changes individually on the overall sentiment analysis performance.

	SentiWordNet	MPQA	Thelwall-Lexicon	Average
Words found in the lexicon	54.86	16.81	9.61	27.10
Hidden words	45.14	83.19	90.39	72.90
Words flipped their sentiment orientation	65.35	61.29	53.05	59.90
Words changed their sentiment strength	29.30	36.03	46.95	37.43
New opinionated words	49.03	32.89	34.88	38.93

Table 3. Average percentage of words in three datasets, which their sentiment orientation or strength were updated by their SentiCircles

Our evaluation results showed that our SentiCircle representation coupled with the MPQA or Thelwall lexicons gives the highest performance amongst the other three lexicons. However, Table 3 shows that only 9.61% of the words in the three datasets were covered by the Thelwall-Lexicon, and 16.81% by MPQA. Nevertheless, SentiCircle performed best with these two lexicons, which suggests that it was able to cope with this low coverage by assigning sentiment to a large proportion of the *hidden* words.

6.2 Incorporating Conceptual Semantics in Sentiment Detection

In this section we report the results when enriching the SentiCircle representation with conceptual semantics by using the augmentation method (see Section 3.3). As explained earlier, we used AlchemyAPI to extract the semantic concepts for the three evaluation datasets. Table 4 lists the total number of entities and concepts extracted for each dataset.

	HCR	OMD	STS-Gold
No. of Entities	1194	1392	2735
No. of Concepts	14	19	23

Table 4. Entity/concept extraction statistics of OMD, HCR and STS-Gold using AlchemyAPI.

Figure 6 depicts the win/loss in accuracy and F-measure when adding the conceptual semantics to the SentiCircle model across all datasets. Note that here we used the Thelwall-Lexicon to obtain the word prior sentiments in our three sentiment detection methods.

The results show that the impact of conceptual semantics on the performance of SentiCircles varies across datasets. On the HCR dataset, a 1.21% gain in F-measure is obtained using the Median method. Also the Median method is more affected by semantic incorporation than the Pivot method, where a much clearer shift in performance

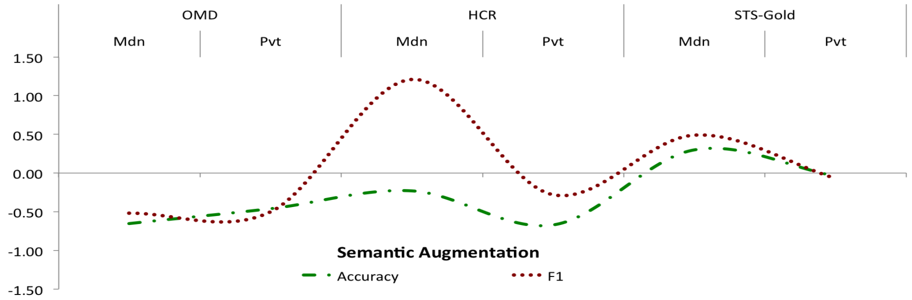


Fig. 6. Win/Loss in Accuracy and F-measure of incorporating conceptual semantics into SentiCircles, where Mdn: SentiCircle with Median method, Pvt: SentiCircle with Pivot method.

can be observed across datasets. This can be explained as the Median method considers all the incorporated concepts in the SentiCircle, whereas the Pivot method focus more on concepts that are associated to target terms in tweets (See Section 4).

As shown in Table 4, the number of entities extracted for the STS-Gold dataset is almost twice as for HCR or OMD. Nonetheless, the results show that semantic incorporation seems to have a lower impact on the STS-Gold dataset than on the OMD and HCR datasets. This might be due to the topical-focus of each dataset. While OMD and HCR are both composed of a smaller number of tweets about specific topics (the US Health Care Reform bill and the Obama-McCain debate), the STS-Gold dataset contains a larger number of tweets with no particular topical focus.

7 Discussion and Future Work

We showed the potential of using SentiCircle for sentiment detection of tweets. The evaluation was performed on three Twitter datasets and using three different sentiment lexicons. Compared to SentiStrength, the results were not as conclusive, since SentiStrength slightly outperformed SentiCircle on the STS-Gold dataset, and also yielded marginally better F-measure for the HCR dataset. This might be due to the different topic distribution in the datasets. STS-Gold dataset contains random tweets, with no particular topic focus, whereas OMD and HCR consist of tweets that discuss specific topics, and thus the contextual semantics extracted by SentiCircle are probably more representative in these datasets than in STS-Gold. Other important characteristics could be the sparseness degree of data and the positive and negative distribution of tweets. In future work, we plan to further investigate these issues and their individual influence on the performance of our approach.

SentiCircle updates the sentiment of terms to match their context. Part of our future work is to study which type of terms change their sentiment, and which are more stable. This can help improving performance by filtering out stable terms. Another evaluation dimension is how SentiCircle performs in monitoring sentiment around a subject over time, to further demonstrate its power and value of updating terms' sentiment with time.

Since all the baselines used in our evaluation are purely syntactical methods, we aim in the future to compare our approach to other, which take word semantics into account for sentiment detection, such as SenticNet.

In this work, the context, in which the semantics of words were extracted and used for sentiment, is defined at the corpus level, that is, by taking into account the occurrence patterns of terms in the whole tweet corpus. We are currently investigating defining the context of terms at more fine-grained levels including tweet- and sentence levels.

We proposed and tested methods that assign positive, negative or neutral sentiment to terms and tweets based on their corresponding SentiCircle representations. However, there could be a need for cases where terms with "Mixed" sentiment emerge, when their SentiCircle representations consist of positive and negative terms only.

We extracted opinion targets (Pivot terms) in the Pivot-Method simply by looking at their POS-tags assuming that all pivot terms in a given tweet receive similar sentiment. We aim next to evaluate this process and to consider cases, where the tweet contains several pivot terms of different sentiment orientations.

We investigated adding conceptual semantics to SentiCircles and studied their impact on the overall sentiment detection performance. In general, a marginal loss in performance (especially in accuracy) was observed comparing to only using contextual semantics. This might be due to the generality of some of the extracted concepts (e.g., "person", "company"), which were applied to many terms of opposite sentiment. These concepts were regarded as normal terms in tweets, and had their own SentiCircles, which might have had a negative impact on the extraction of sentiment. A fix might be to extract more specific concepts, using other concept extractors (e.g., SenticNet).

8 Conclusions

In this paper we proposed a novel semantic sentiment approach called SentiCircle, which captures the semantics of words from their context and update their sentiment orientations and strengths accordingly. We described the use of SentiCircle for lexicon-based sentiment identification of tweets using different methods. We showed that our approach outperformed other lexicon labelling methods and overtake the state-of-the-art SentiStrength approach in accuracy, with a marginal drop in F-measure. Unlike most other lexicon-based approaches, SentiCircle was able to update the sentiment strength of many terms dynamically based on their contextual semantics.

We enriched the SentiCircle representation with conceptual semantics extracted using AlchemyAPI. Results showed that adding concepts to SentiCircle has a good potential, and indicated that the use of conceptual semantics with SentiCircle might be more appropriate when the datasets being analysed are large and cover a wide range of topics.

Acknowledgment

This work was supported by the EU-FP7 project SENSE4US (grant no. 611242).

References

1. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proc. ACL 2011 Workshop on Languages in Social Media. Portland, Oregon (2011)
2. Aue, A., Gamon, M.: Customizing sentiment classifiers to new domains: A case study. In: Proc. recent advances in natural language processing (RANLP). Borovets, Bulgaria (2005)
3. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Seventh conference on International Language Resources and Evaluation, Malta. Retrieved May. Valletta, Malta (2010)

4. Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: Proceedings of COLING. Beijing, China (2010)
5. Bifet, A., Frank, E.: Sentiment knowledge discovery in twitter streaming data. In: Discovery Science. Canberra, Australia (2010)
6. Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. Journal of Computational Science 2(1), 1–8 (2011)
7. Cambria, E.: An introduction to concept-level sentiment analysis. In: Advances in Soft Computing and Its Applications, pp. 478–483. Springer (2013)
8. Cambria, E., Havasi, C., Hussain, A.: Senticnet 2: A semantic and affective resource for opinion mining and sentiment analysis. In: FLAIRS Conference. pp. 202–207 (2012)
9. Diakopoulos, N., Shamma, D.: Characterizing debate performance via aggregated twitter sentiment. In: Proc. 28th Int. Conf. on Human factors in computing systems. ACM (2010)
10. Firth, J.R.: A synopsis of linguistic theory. Studies in Linguistic Analysis (1930- 1955)
11. Garcia-Moya, L., Anaya-Sanchez, H., Berlanga-Llavori, R.: Retrieving product features and opinions from customer reviews. IEEE Intelligent Systems 28(3), 19–27 (2013)
12. Kouloumpis, E., Wilson, T., Moore, J.: Twitter sentiment analysis: The good the bad and the omg! In: Proceedings of the ICWSM. Barcelona, Spain (2011)
13. Liu, B.: Sentiment analysis and subjectivity. Taylor and Francis Group (2010)
14. O'Connor, B., Balasubramanyan, R., Routledge, B.R., Smith, N.A.: From tweets to polls: Linking text sentiment to public opinion time series. ICWSM 11, 122–129 (2010)
15. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of LREC 2010. Valletta, Malta (2010)
16. Saif, H., Fernandez, M., He, Y., Alani, H.: Evaluation datasets for twitter sentiment analysis. In: Proceedings, 1st Workshop on Emotion and Sentiment in Social and Expressive Media (ESSEM) in conjunction with AI*IA Conference. Turin, Italy (2013)
17. Saif, H., He, Y., Alani, H.: Alleviating data sparsity for twitter sentiment analysis. In: Proc. Workshop on Making Sense of Microposts (#MSM2012) in WWW 2012. Lyon, France (2012)
18. Saif, H., He, Y., Alani, H.: Semantic sentiment analysis of twitter. In: Proc. 11th Int. Semantic Web Conf. (ISWC). Boston, MA (2012)
19. Speriosu, M., Sudan, N., Upadhyay, S., Baldridge, J.: Twitter polarity classification with label propagation over lexical links and the follower graph. In: Proceedings of the EMNLP First workshop on Unsupervised Learning in NLP. Edinburgh, Scotland (2011)
20. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational linguistics 37(2), 267–307 (2011)
21. Takamura, H., Inui, T., Okumura, M.: Extracting semantic orientations of words using spin model. In: Proc. 43rd Annual Meeting on Association for Computational Linguistics (2005)
22. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. J. American Society for Information Science and Technology 63(1), 163–173 (2012)
23. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. J. American Society for Info. Science and Technology 61(12) (2010)
24. Turney, P., Littman, M.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems 21, 315–346 (2003)
25. Turney, P.D., Pantel, P., et al.: From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research 37(1), 141–188 (2010)
26. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Vancouver, British Columbia, Canada (2005)
27. Wittgenstein, L.: Philosophical Investigations. Blackwell, London, UK (1953, 2001)