Information Processing and Management 52 (2016) 5–19

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Contextual semantics for sentiment analysis of Twitter

Hassan Saif[a,*], Yulan He[b], Miriam Fernandez[a], Harith Alani[a]

[a] Knowledge Media Institute, The Open University, United Kingdom
[b] School of Engineering and Applied Science, Aston University, United Kingdom

ARTICLE INFO

Article history:
Received 30 April 2014
Revised 24 November 2014
Accepted 28 January 2015
Available online 7 March 2015

Keywords:
Sentiment analysis
Contextual semantics
Twitter

ABSTRACT

Sentiment analysis on Twitter has attracted much attention recently due to its wide applications in both, commercial and public sectors. In this paper we present SentiCircles, a lexicon-based approach for sentiment analysis on Twitter. Different from typical lexicon-based approaches, which offer a fixed and static prior sentiment polarities of words regardless of their context, SentiCircles takes into account the co-occurrence patterns of words in different contexts in tweets to capture their semantics and update their pre-assigned strength and polarity in sentiment lexicons accordingly. Our approach allows for the detection of sentiment at both entity-level and tweet-level. We evaluate our proposed approach on three Twitter datasets using three different sentiment lexicons to derive word prior sentiments. Results show that our approach significantly outperforms the baselines in accuracy and *F*-measure for entity-level subjectivity (neutral vs. polar) and polarity (positive vs. negative) detections. For tweet-level sentiment detection, our approach performs better than the state-of-the-art SentiStrength by 4–5% in accuracy in two datasets, but falls marginally behind by 1% in *F*-measure in the third dataset.

1. Introduction

Twitter sentiment analysis has attracted much attention due to the rapid growth in Twitter's popularity as a platform for people to express their opinions and attitudes towards a great variety of topics. Approaches to Twitter sentiment analysis tend to focus on the identification of sentiment of individual tweets (*tweet-level sentiment detection*). Broadly speaking, existing work on tweet-level sentiment detection follows two main types of approaches, supervised learning or lexicon-based.

Supervised learning approaches require training data for sentiment classifier learning. In Twitter, training data are typically obtained by either assuming that tweets' polarities (positive, negative, neutral) can be inferred using emoticons (Go, Bhayani, & Huang, 2009; Kouloumpis, Wilson, & Moore, 2011; Pak & Paroubek, 2010; Saif, He, & Alani, 2012b) or by taking consensus from the results returned by the sentiment detection websites (Barbosa & Feng, 2010). Moreover, supervised approaches are domain-dependent and require re-training with the arrival of new data (Aue & Gamon, 2005). Given the great variety of topics that constantly emerge from Twitter, these limitations affect the applicability of such approaches.

On the other hand, lexicon-based approaches do not require training data. Instead, they use lexicons of words weighted with their sentiment orientations to determine the overall sentiment of a given text. These approaches have shown to work effectively on conventional text (Liu, 2010). However, traditional lexicons tend to be ill-suited for Twitter data, which often

[*] Corresponding author.
E-mail addresses: h.saif@open.ac.uk (H. Saif), y.he@cantab.net (Y. He), m.fernandez@open.ac.uk (M. Fernandez), h.alani@open.ac.uk (H. Alani).

http://dx.doi.org/10.1016/j.ipm.2015.01.005
0306-4573/© 2015 Elsevier B.V. All rights reserved.

contains a large number of malformed words and colloquial expressions (e.g., “looov”, “luv”, “gr8”). Moreover, many lexicon-based approaches also make use of the lexical structure of a sentence to determine its sentiment, which becomes problematic in Twitter, where ungrammatical sentences are very common due to the 140-character length limit. Aiming to overcome these limitations, Thelwall, Buckley, Paltoglou, Cai, and Kappas (2010) and Thelwall, Buckley, and Paltoglou (2012) introduced a human-coded lexicon of words and phrases specifically built to work with social data. They proposed an algorithm called *SentiStrength* that utilises the lexicon to identify the sentiment strength of informal text (e.g., tweets, status updates). We refer to this lexicon as Thelwall-Lexicon hereafter.

SentiStrength has received much attention in recent years due to its relatively good and consistent performance on social media data. Nevertheless, similarly to other lexicon-based approaches, SentiStrength and its underlying Thelwall-Lexicon face two main limitations. Firstly, SentiStrength is confined with the fixed set of words that appear in the Thelwall-Lexicon. Words that do not appear in the lexicon are often not considered when analysing sentiment (Liu, 2010; Xu, Peng, & Cheng, 2012), which may create a problem when dealing with Twitter data, where new expressions and jargons constantly emerge. Secondly and more importantly, SentiStregnth and the like offer fixed, context-independent, word-sentiment orientations and strengths. For example, SentiStrength assigns the same sentiment strength to the word “good” in “`It is a very good phone indeed!`” and in “`I will leave you for good this time!`”. Although a training algorithm has been proposed to optimise the terms' sentiment scores in Thelwall-Lexicon (Thelwall et al., 2010), it requires frequent retraining from human-coded data, which is labour-intensive and domain dependent.

In this paper we introduce an approach called SentiCircles (Saif, Fernandez, He, & Alani, 2014b), which builds a dynamic representation of words that captures their contextual semantics (i.e., semantics inferred from the co-occurrence patterns of words in text) in order to tune their pre-assigned sentiment strength and polarity in a given sentiment lexicon.

Contextual semantics (aka statistical semantics) (Wittgenstein, 1953) has been traditionally used in diverse areas of computer science, including Natural Language Processing and Information Retrieval (Turney & Pantel, 2010). The main principle behind the notion of contextual semantics comes from the dictum – “You shall know a word by the company it keeps!” (Firth, 1930–1955). 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 assess the performance of our proposed SentiCircle approach in two different sentiment analysis tasks: (i) *entity-level sentiment* detection, which detects sentiment towards a particular entity or topic (e.g., Obama, Microsoft, iPad) and (ii) *tweet-level sentiment* detection, which identifies the overall sentiment of *individual* tweets. To this end, we propose three different methods, which utilise several trigonometric identities on the SentiCircle representation to perform both sentiment analysis tasks.

We evaluate and test our approach under different settings (three different sentiment lexicons and three different datasets) and compare its performance against various lexicon baseline methods. We also compare our approach against SentiStrength, which, to our knowledge, is considered one of the best lexicon-based sentiment detection approaches for social media. For entity-level sentiment detection, our experimental results show that our proposed approach, based on SentiCircles, outperforms all the other methods by nearly 20% in accuracy and 30–40% in *F*-measure for subjectivity detection (neutral vs. polar). For tweet-level sentiment detection, our approach outperforms SentiStrength by 4–5% in accuracy in two datasets, but falls marginally behind by 1% in *F*-measure on the third dataset.

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 latent semantics of words from their co-occurrence patterns and update their sentiment orientations accordingly.
- Propose three different methods of employing SentiCircles for tweet-level sentiment detection.
- Conduct a series of experiments and test the effectiveness of our proposed approach for both entity- and tweet-level sentiment detection against several baselines, including SentiStrength.
- Perform a runtime analysis of our approach to demonstrate its scalability.
- Build and release the STS-Gold (Saif, Fernandez, He, & Alani, 2013), a new gold-standard dataset that allows for evaluating both, tweet- and entity-level sentiment analysis approaches.

The remainder of this paper is structured as follows. Related work on tweet-level and entity-level sentiment analysis is discussed in Section 2. The proposed SentiCircle representation of words is presented in Section 3. How to apply SentiCircles for sentiment analysis is described in Section 4. Experimental setup 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 as positive or negative. They can be categorised as *supervised methods* (those which need training data) and *lexicon-based methods* (those based on dictionaries of terms with associated sentiment orientations).

2.1. Supervised machine learning methods

Supervised methods are based on training classifiers, such as Naïve Bayes (NB), Maximum Entropy (MaxEnt), and Support Vector Machines (SVMs), from various combinations of features, such as word *n*-grams (Bifet & Frank, 2010; Go et al., 2009; Pak & Paroubek, 2010), Part-Of-Speech (POS) tags (Agarwal, Xie, Vovsha, Rambow, & Passonneau, 2011; Barbosa & Feng, 2010) with or without words' prior sentiment, words' semantic concepts (Saif et al., 2012b), sentiment-topic features (Saif, He, & Alani, 2012a), semantic patterns (Saif, He, Fernandez, & Alani, 2014) and tweets syntax features (e.g., hashtags, retweets, punctuations, etc.) (Kouloumpis et al., 2011). These methods have achieved relatively good results with accuracies reported in the range of 80–84% as in Saif et al. (2012a, 2012b). However, training data are difficult to obtain (Liu, 2010), especially for the continuously changing and evolving Twitter data. Aiming to overcome this limitation, the distance supervision approach (Go et al., 2009) makes use of automatically generated training data, where emoticons such as ":-)" and ":(" are typically used to label tweets as positive or negative. However, automatic labelling of training data introduces errors that may affect the performance of the classifiers (Speriosu, Sudan, Upadhyay, & Baldridge, 2011). Another limitation of supervised approaches is their domain dependence, i.e., classifiers trained on data from one domain (e.g., tweets relating to health reform) produce unsatisfactory performance when applied to data from a different domain (e.g., tweets relating to products) (Aue & Gamon, 2005).

2.2. Lexicon-based methods

Lexicon-based methods try to overcome the aforementioned limitations by using the sentiment orientation of words and phrases in a given document to calculate its overall sentiment. Instead of using training data, lexicon-based methods rely on sentiment lexicons, i.e. pre-built dictionaries of words with associated sentiment orientations such as SentiWordNet (Baccianella, Esuli, & Sebastiani, 2010), MPQA subjectivity lexicon (Wilson, Wiebe, & Hoffmann, 2005), or the LIWC lexicon (Pennebaker, Mehl, & Niederhoffer, 2003). These lexicons, although costly to obtain, once constructed they are applicable to a wide variety of domains. To reduce the cost of building these sentiment lexicons, some approaches apply bootstrapping techniques to add words to an initial subset or seeds (Andreevskaia & Bergler, 2006; Neviarouskaya, Prendinger, & Ishizuka, 2011).

Thelwall et al. (2010, 2012) proposed SentiStrength, a lexicon-based method for sentiment detection on the Social Web. SentiStrength overcomes the problem of ill-formed language by applying several lexical rules, such as the existence of emoticons, intensifiers, negation and booster words (e.g., absolutely, extremely), to compute the average sentiment strength of an online post. Note that this method, as well as other existing lexicon-based methods (Taboada, Brooke, Tofiloski, Voll, & Stede, 2011), do not only focus on polarity (positive/negative sentiment) detection, but also on identifying sentiment strength. In the case of Taboada et al. (2011) sentiment strength varies between −5 (very negative) to +5 (very positive).

One limitation of the lexicon-based methods is that they are restricted by their lexicons, and more particularly, by the use of static prior sentiment values of terms regardless of their contexts. Although authors in Thelwall et al. (2010) have proposed an algorithm to update the sentiment strength assigned to the terms in the lexicon, this algorithm requires to be trained 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 (Cambria, 2013).

2.3. Semantic sentiment methods

Several methods have been proposed for exploring semantics for sentiment analysis, which can be categorised into *contextual semantic*, and *conceptual semantic* approaches.

Contextual semantic approaches determine semantics from the co-occurrence patterns of words, also known as *statistical semantics* (Turney & Pantel, 2010; Wittgenstein, 1953), and have often been used for sentiment analysis (Takamura, Inui, & Okumura, 2005; Turney, 2002; Turney & Littman, 2003). Turney and Littman (2003), for example, used *pointwise mutual information* (PMI) to measure the statistical correlation between a given word and a balanced set of 14 positive and negative paradigm words (e.g., good, nice, nasty, poor). The word has positive orientation if it has a stronger degree of association to positive words than to negative ones, and vice versa. Although this work does not require large lexical input knowledge, its identification speed is very limited (Xu et al., 2012) because it uses web search engines in order to retrieve the relative co-occurrence frequencies of words. More importantly, this approach is unable to assign sentiment to words with domain specific orientations (Ding, Liu, & Yu, 2008) due to its limited choice of paradigm words and its use of the entire web as a corpus. For example, it is unable to distinguish between "`Heavy`" as a negative word when describing a mobile phone and as positive word when describing a wood dining table.

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 (Saif et al., 2012b). SenticNet (Cambria, Havasi, & Hussain, 2012)[1] 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) (Garcia-Moya, Anaya-Sanchez, & Berlanga-Llavori, 2013). Unlike SentiStrength (Thelwall et al., 2010), SenticNet is not tailored for Twitter and the like. Although conceptual semantic approaches have been shown to outperform purely syntactical approaches (Cambria, 2013), 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.

2.4. *Entity-level sentiment analysis approaches*

Compared to tweet-level sentiment analysis, there has been relatively less research on *entity-level sentiment analysis*. Batra and Rao (2010), propose the use of probabilistic models measuring the sentiment of an entity as an aggregation of the sentiment of all tweets that are associated with that entity. However, the sentiment of a tweet may or may not be related to the sentiment of an entity which appears in it. For example, the tweet, "`The new Twitter for iPhone is awesome.`", expresses a positive sentiment towards "`Twitter`", but not towards "`iPhone`". More recently, *supervised approaches* have emerged attempting to address the problem of entity sentiment detection. Jiang, Yu, Zhou, Liu, and Zhao (2011) proposed to train SVM binary classifiers on nearly 2000 manually annotated tweets about 5 chosen entities {`Obama, Google, iPad, Lakers, Lady Gaga`}, achieving about 68% accuracy for entity-level sentiment classification. Meng et al. (2012) also trained a SVM classifier on 500 annotated tweets about 6 entities {`Obama, Lady Gaga, David Cameron, Nokia, Apple, Microsoft`}. The trained classifier achieved 83% *F*-measure on entity-level sentiment classification based on 5-fold cross validation using the annotated tweets. However, these approaches rely on training data, which is costly to obtain and lack portability to other domains.

Aiming at addressing the limitations of the aforementioned works, we have designed our SentiCircle approach in a way that: (1) it follows a lexicon-based approach and hence it can be applied to Twitter data from different domains, (2) it is context-sensitive, i.e., it updates the sentiment score of words on the fly based on the contexts they appear in (i.e., their contextual semantics), and (3) it works for entity- and tweet-level sentiment detection.

3. SentiCircle representation of words

In this section, we introduce our SentiCircle approach and its use for capturing the contextual semantics and sentiment of words.

SentiCircle aims to learn the sentiment orientation of words from their contextual semantics. The main notion behind this 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 words' contextual semantics we follow the distributional hypothesis that, words that occur in similar contexts tend to have similar meanings (Turney & Pantel, 2010; Wittgenstein, 1953). Therefore, the contextual semantics of a term m in our approach is computed from its co-occurrence patterns with other terms.

Fig. 1 shows the systematic workflow of our approach, which can be summarised in the following steps:

- Term Indexing: This step creates an index of terms (term-index) from a collection of tweet messages. Several text processing procedures are applied during the process such as: Filtering Non-English terms, Part-Of-Speech tagging and Negation (Section 3.2). Note that we do not remove stopwords from tweets since they tend to carry sentiment information as shown in Saif, Fernandez, He, and Alani (2014a) and Saif, Fernandez, and Alani (2014).
- Term-Context Vector Generation: This step represents each term m as a vector of all its context terms (i.e., terms that occur with m in the same context) in the tweets. (See below for a formal definition.)
- Contextual Features Generation: We compute, for each term, its degree of correlation to all its context terms. We also assign an initial score to these context terms using an external sentiment lexicon.
- SentiCircle Generation: This step converts the term-context vector of m into a 2D geometric circle, which is composed of points denoting the context terms of m. Each context term is located in the circle based on its angle (defined by its prior sentiment), and its radius (defined by its degree of correlation with the term m) (Section 3.1).
- Sentiment Identification: Here, we apply different methods that utilise several trigonometric identities on SentiCircles to perform sentiment identification at either entity- or tweet-level (Section 4).

Definition (Term-Context Vector)**.** Given a set of tweet messages $\mathcal{T}$, the term-context vector of a term m is a vector $\vec{c} = (c_1, c_2, \ldots, c_n)$ of terms that occur with m in any tweet in $\mathcal{T}$.

The contextual semantics of m is determined by its semantic relation to each context term $c_i \in \vec{c}$. We compute the individual semantic relation between m and a context term c_i, by assigning the following two main features to c_i:

[1] http://sentic.net/.

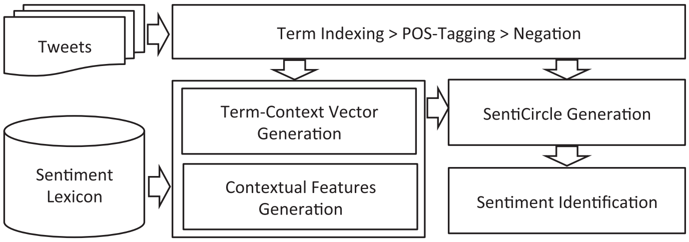

Tweets
Term Indexing > POS-Tagging > Negation
Term-Context Vector Generation
Contextual Features Generation
SentiCircle Generation
Sentiment Lexicon
Sentiment Identification

Fig. 1. The systematic workflow of the SentiCircle approach for sentiment analysis.

- Prior Sentiment Score: Each context term c_i is assigned to a prior sentiment score based on its POS tag(s) by using one of the three external sentiment lexicons used in this paper (Section 5.3).
- Term Degree of Correlation (TDOC): This feature represents the degree of correlation between a term m and its context term $c_i \in \vec{c}$ (i.e., how important c_i is to m). Inspired by the TF-IDF weighting scheme, we compute the value of this feature as:

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

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.

3.1. SentiCircles representation of semantics

Now we have for each term m a vector of its context terms $\vec{c}$ along with the two semantic mutual features between m and each $c_i \in \vec{c}$. From these information, we represent the contextual semantics of the term m 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:

$$r^2 - 2rr_0 \cos(\theta - \phi) + r_0^2 = a^2 \tag{2}$$

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 context 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 T-DOC value of the term c_i as:

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

Note that since each context term may have several prior sentiment scores based on its POS tag(s) in tweets, the value returned by function $\mathrm{Prior\_Sentiment}(c_i)$ is the average sentiment score of all the term's occurrences.

We normalise the radii of all the 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.

The SentiCircle in the *polar coordinate system* can be divided into four sentiment quadrants as shown in Fig. 2. 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 ($0 \le r_i \le 1$), 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:

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

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

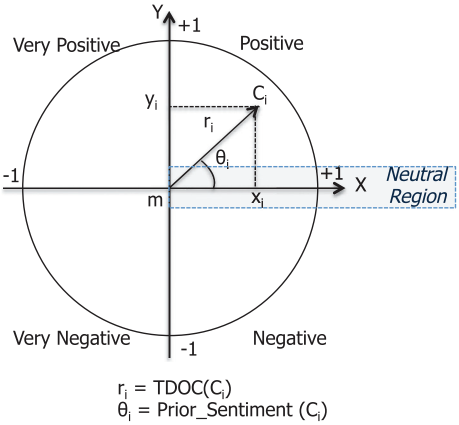


Fig. 2. SentiCircle of a term *m*.

sentiment strength of the term. The smaller the *x* value, the stronger the sentiment.[2] Moreover, a small region called the "*Neutral Region*" can be defined. This region, as shown in Fig. 2, 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.

Fig. 3 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 Fig. 3(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 term degree of correlation) is calculated and represented in the SentiCircle separately by means of the projection of the context term along *X*-axis (sentiment strength) and *Y*-axis (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. Negation

When constructing the SentiCircle representations, if a term *t*, with an associated sentiment score s_t appears in the tweet within the vicinity of a negation, its sentiment score is negated for the construction of the SentiCircle ($-s_t$). For example, in the tweet "`iPad is not amazing!`", the term "`amazing`" is preceded by a negation. Therefore, instead of using its original sentiment score (0.75 in the SentiWordNet lexicon for example) we use this score negated (−0.75). The negation words are collected from the General Inquirer under the NOTLW category.[3]

3.3. Senti-median – contextual sentiment value

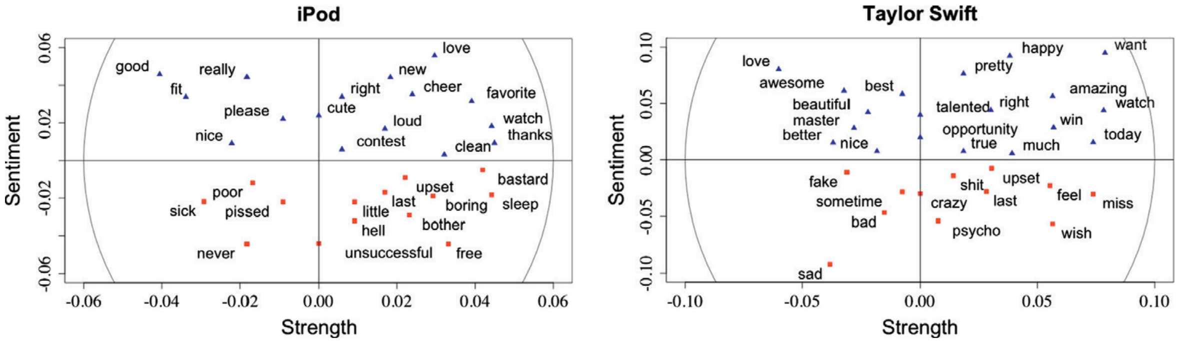
The previous examples in Section 3.1 show that, although we use external lexicons to assign initial sentiment scores to terms, our SentiCircle representation is able to amend these scores according to the context in which each term is used. To compute the new sentiment of the term based on its SentiCircle 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 $(p_1, p_2, \ldots, p_n)$ in a SentiCircle Ω, the 2D geometric median *g* is defined as:

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

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

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



(a) iPod.

(b) Taylor Swift.

Fig. 3. Example SentiCircles for “iPod” and “Taylor Swift”. We have removed points near the origin for easy 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 bearing a negative sentiment.

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.

4. SentiCircles for sentiment analysis

In this section, we show how the SentiCircle representation can be used in two different sentiment analysis tasks; entity- and tweet-level sentiment detection.

4.1. Entity-level sentiment detection

Given an entity, $e_i \in \mathcal{E}$, and its corresponding SentiCircle representation, the sentiment of the entity is given by the Senti-Median g of the SentiCircle (i.e., by the geometric median of all the points that compose the SentiCircle). Following the representation provided in Fig. 2, if the Senti-Median g lies inside the “*Neutral Region*”, the entity will have a **neutral sentiment**. If g lies in one of the positive quadrants, the entity will have a **positive sentiment** and, if g lies in the negative quadrants, the entity will have a **negative sentiment**.

Formally, given a Senti-Median g_e of an entity e, the entity-sentiment function $\mathcal{L}$ works as:

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

where λ is the threshold that defines the Y-axis boundary of the neutral region. Section 5.4 illustrates how this threshold is computed.

4.2. Tweet-level sentiment detection

Given a tweet, $t_i \in \mathcal{T}$, there are several ways in which the SentiCircle representations of the terms that compose the 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 SentiCircle representation. These five SentiCircles can be combined in different ways in order to extract the sentiment associated to the tweet. In this section we propose three different methods that exploit the SentiCircle representation for tweet-level sentiment detection.

4.2.1. The Median method

This method works by representing each tweet message $t_i \in \mathcal{T}$ as a vector of Senti-Medians $\vec{g} = (g_1, g_2, \ldots, g_n)$ 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 to term m_j. Eq. (5) is then used to calculate the median point q of $\vec{g}$, which we use to determine the overall sentiment of tweet t_i using Function 6.

4.2.2. The 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:

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

where $s \in \mathcal{S} = \{Positive, Negative, Neutral\}$ is the sentiment label, $\vec{p}$ is a vector of all pivot terms in a tweet, $N_{\vec{p}}$ and $N_{\vec{w}}$ 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) as shown in Fig. 4. Note that the impact value is doubled for all terms located either in the "*Very Positive*" or in the "*Very Negative*" quadrants.

4.2.3. *The Pivot-Hybrid method*

The Pivot method, as described in the previous section, relies on both the syntactical structure of a tweet and the sentiment relations among its terms. As such, it may suffer from the lack of pivot terms when the tweet message is too short or it contains many ill-formed words. In such a case, we resort to the Median method, and call this the `Pivot-Hybrid` 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. This enables us to assess the influence of different corpora and lexicons on the performance of our SentiCircle approach.

5.1. *Datasets*

In this section, we present the three datasets used for the evaluation; OMD, HCR and STS-Gold. We use the OMD (Diakopoulos & Shamma, 2010) and HCR (Speriosu et al., 2011) datasets[4] to assess the performance of our approach at the tweet level only since they provide human annotations for tweets but not for entities (i.e., each tweet is assigned a positive, negative or neutral sentiment label).

Due to the lack of gold-standard datasets for evaluating entity-level sentiment, we have generated an additional dataset (STS-Gold) (Saif et al., 2013).[5] This dataset contains both, tweet and entity sentiment ratings and therefore, we use it in this paper to assess the performance of SentiCircles at both the entity and the tweet levels.

Numbers of positive and negative tweets within the three datasets are summarised in Table 1 and further described below:

5.1.1. *Obama-McCain Debate (OMD)*

This dataset was constructed from 3238 tweets crawled during the first U.S. presidential TV debate in September 2008 (Diakopoulos & Shamma, 2010). Sentiment ratings of these tweets were acquired using Amazon Mechanical Turk, where each tweet was rated by one or more voter as either positive, negative, mixed, or other. We only keep those tweets rated by at least three voters with two-third of the votes being either positive or negative to ensure their sentiment polarity. This resulted in a set of 1,081 tweets with 393 positive and 688 negative ones.

5.1.2. *Health Care Reform (HCR)*

The HCR dataset was built by crawling tweets containing the hashtag "#hcr" (health care reform) in March 2010 [14]. A subset of this corpus was manually annotated with three polarity labels (positive, negative, neutral) and split into training and test sets. In this paper we focus on identifying positive and negative tweets and therefore we exclude neutral tweets from this dataset. This resulted in a set of 1354 tweets, 397 positive and 957 negative.

5.1.3. *Standford Sentiment Gold Standard (STS-Gold)*

We constructed this dataset as a subset of the Stanford Twitter Sentiment Corpus (STS) (Go et al., 2009). It contains 2,034 tweets (632 positive and 1402 negative) and 58 entities manually annotated by three different human evaluators (see

[4] https://bitbucket.org/speriosu/updown.

[5] http://tweenator.com.

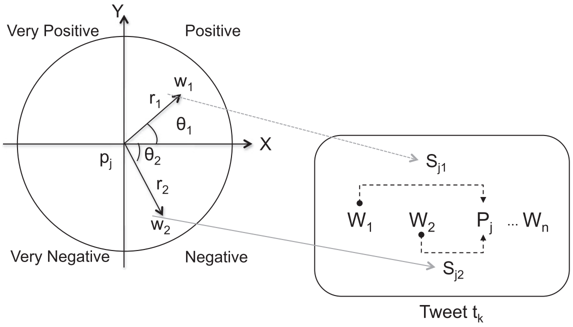


Fig. 4. The Pivot method. The figure shows a SentiCircle of a pivot term p_i. The sentiment strength S_{i1} of word w_1 with respective to the pivot term p_i is the radius r_1 in the SentiCircle, and likewise for S_{i2}.

Table 1
Twitter datasets used for the evaluation.

Dataset	Tweets	Positive	Negative
OMD Diakopoulos and Shamma (2010)	1081	393	688
HCR Speriosu et al. (2011)	1354	397	957
STS-Gold Saif et al. (2013)	2034	632	1402

Table 3). To avoid noisy or misleading data in the created dataset, the entities and tweets selected for these dataset are those for which the three human evaluators agreed on the same sentiment label.

In the following we describe the construction and the annotation of the STS-Gold dataset.

Data acquisition: To construct this dataset, we first extracted all named entities from a collection of 180 K tweets randomly selected from the original Stanford Twitter corpus (Go et al., 2009). To this end, we used AlchemyAPI,[6] an online service that allows for the extraction of entities from text along with their associated semantic concept class (e.g., Person, Company, City). After that, we identified the top most frequent semantic concepts and, selected under each of them, the top 2 most frequent and 2 mid-frequent entities. For example, for the semantic concept *Person* we selected the top most frequent entities (Taylor Swift and Obama) as well as two mid frequent entities (Oprah and Lebron). This resulted in 28 different entities along with their 7 associated concepts as shown in Table 2.

The next step was to construct and prepare a collection of tweets for sentiment annotation, ensuring that each tweet in the collection contains one or more of the 28 entities listed in Table 2. To this aim, we randomly selected 100 tweets from the remaining part of the STS corpus for each of the 28 entities, i.e., a total of 2800 tweets. We further added another 200 tweets without specific reference to any entities to add up a total of 3000 tweets. Afterwards, we applied AlchemyAPI on the selected 3000 tweets. Apart from the initial 28 entities the extraction tool returned 119 additional entities, providing a total of 147 entities for the 3000 selected tweets.

Data annotation: We asked three graduate students to manually label each of the 3000 tweets with one of the five classes: (`Negative, Positive, Neutral, Mixed` and `Other`). The "`Mixed`" label was assigned to tweets containing mixed sentiment and "`Other`" to those that were difficult to decide on a proper label. The students were also asked to annotate each entity contained in a tweet with the same five sentiment classes. The students were provided with a booklet explaining both the tweet-level and the entity-level annotation tasks. The booklet also contains a list of key instructions (Saif et al., 2013). It is worth noting that the annotation was done using Tweenator, 5 an online tool that we previously built to annotate tweet messages (Saif et al., 2012a).

We measured the inter-annotation agreement using the Krippendorff's alpha metric (Krippendorff, 2004, chap. 11), obtaining an agreement of $\alpha_t = 0.765$ for the tweet-level annotation task. For the entity-level annotation task, if we measured sentiment of entity for each individual tweet, we only obtained $\alpha_e = 0.416$ which is relatively low for the annotated data to be used. However, if we measured the aggregated sentiment for each entity, we got a very high inter-annotator agreement of $\alpha_e = 0.964$.

To construct the final STS-Gold dataset we selected those tweets and entities for which our three annotators agreed on the sentiment labels, discarding any possible noisy data from the constructed dataset. As shown in Table 3 the STS-Gold dataset contains 13 negative, 27 positive and 18 neutral entities as well as 1402 negative, 632 positive and 77 neutral tweets.

It it worth noting that we use SentiCircles to perform polarity classification of tweets, and therefore we only consider those positive and negative tweets in the STS-Gold dataset for evaluation.

[6] www.alchemyapi.com.

Table 2
28 Entities, with their semantic concepts, used to build STS-Gold.

Concept	Top 2 entities	Mid 2 entities
Person	Taylor Swift, Obama	Oprah, Lebron
Company	Facebook, Youtube	Starbucks, McDonalds
City	London, Vegas	Sydney, Seattle
Country	England, US	Brazil, Scotland
Organisation	Lakers, Cavs	Nasa, UN
Technology	iPhone, iPod	Xbox, PSP
HealthCondition	Headache, Flu	Cancer, Fever

Table 3
Number of tweets and entities under each class.

Class	Negative	Positive	Neutral	Mixed	Other
No. of entities	13	27	18	–	–
No. of tweets	1402	632	77	90	4

5.2. Sentiment lexicons

As describe in Section 3, the initial sentiment of terms in a 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 (Baccianella et al., 2010), (ii) the MPQA subjectivity lexicon (Wilson et al., 2005), and (iii) Thelwall-Lexicon (Thelwall et al., 2010, 2012).

5.3. Baselines

We compare the performance of our propose SentiCircle representation when being used for tweet and entity sentiment analysis against the following baselines:

Lexicon labelling method. This method uses the MPQA and the SentiWordNet lexicons to extract the sentiment of a given text. If a tweet contains more positive words than negative ones, it is labelled as positive, and vice versa. For entity-level sentiment detection, the sentiment label of an entity is assigned based on the number of positive and negative words that co-occur with the entity in its associated tweets. In our evaluation, we refer to the method that uses the MPQA lexicon as *MPQA-Method* and to the method that uses the SentiWordNet lexicon as *SentiWordNet-Method*.

SentiStrength. SentiStength (Thelwall et al., 2010; Thelwall et al., 2012) is a state-of-the-art lexicon-based sentiment detection approach. It 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). To use SentiStrength for tweet-level sentiment detection, a tweet is considered positive if its positive sentiment strength is 1.5 times higher than the negative one, and negative otherwise.[7] For entity-level sentiment detection, the sentiment of an entity is assigned based on the total number of positive, negative tweets in which the entity occurs. It is worth noticing that SentiStrength requires the 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.

5.4. Thresholds and parameters tuning

When computing the sentiment of a point within a SentiCircle (Function 6) it is necessary to determine beforehand the geometric boundaries of the neutral region (the region defining the neutral terms) within the SentiCircle. While the boundaries of the neutral region are fixed for the X-axis $[0, 1]$ (see Section 3.1), the boundaries of the Y-axis need to be determined. We have observed that the neutral area of a SentiCircle is defined by a high density of terms, since the number of neutral terms in the SentiCircle 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 area delimited by an increase/decrease in the density of terms.

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

Fig. 5 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 the corpus and sentiment lexicon. For evaluation, we have computed nine different neutral regions, one for each corpus and sentiment lexicon used, as shown in Table 5.

6. Experiment results

We report the performance of our proposed approaches in comparison with the baselines in both the entity- and tweet-level sentiment detection tasks. For entity-level sentiment detection, we conduct experiments on the STS-Gold dataset, while for tweet-level sentiment detection, we use the OMD, HCR and STS-Gold datasets.

6.1. Entity-level sentiment detection

For entity-level sentiment detection, we employ our proposed Senti-Median method (see Section 4.1) with SentiWordNet, MPQA and Thelwall lexicons to identify the overall sentiment of the SentiCircle of a given entity. We report the results in accuracy, precision, recall and *F*-measure on two identification tasks: **subjectivity detection**, which identifies whether a given entity is subjective (positive or negative) or objective (neutral). The second task is **polarity detection**, which identifies whether the entity has positive or negative sentiment. Both identification tasks are applied on 58 different entities (see Section 5.1).

It can be observed from the upper panel of Table 4 that, for subjectivity identification, our proposed Senti-Median method outperforms all the baselines with a large margin. In particular, merely using MPQA or SentiWordNet for sentiment labelling fails to detect any objective (neutral) entities. SentiStrength only achieves an *F*-measure of 15% for objective entity detection. On the contrary, our proposed Senti-Median method gives relatively balanced results on both subjective and objective identification. Senti-Median, with the word prior sentiment obtained from SentiWordNet, attains the best overall result with 81% in accuracy and 80% in *F*-measure, which outperforms the baselines by nearly 20% in accuracy and 30–40% in *F*-measure.

The lower panel of Table 4 shows the results of binary polarity identification (positive vs. negative) at entity-level. SentiStrength performs better than MPQA-Method and SentiWordNet-Method, with 85% in accuracy and 84% in *F*-measure. Although our Senti-Median method, with word prior sentiment obtained from either MPQA or Thelwall-Lexicon, does not seem to bring any improvement over SentiStrength, Senti-Median based on SentiWordNet outperforms SentiStrength by 2.5% in accuracy and 1.5% in *F*-measure.

6.2. Tweet-level sentiment detection

For tweet-level sentiment detection, we report the evaluation results using the Median method, the Pivot method and the Pivot-Hybrid method with SentiWordNet, MPQA and Thelwall lexicons on OMD, HCR and STS-Gold datasets. We also compare these results with those obtained from the baselines described in Section 5.3.

Fig. 6 shows the results in accuracy (left column) and average *F*-measure (right column) of all the methods and across all the datasets. It can be observed that all our three methods outperform the MPQA-Method and SentiWordNet-Method

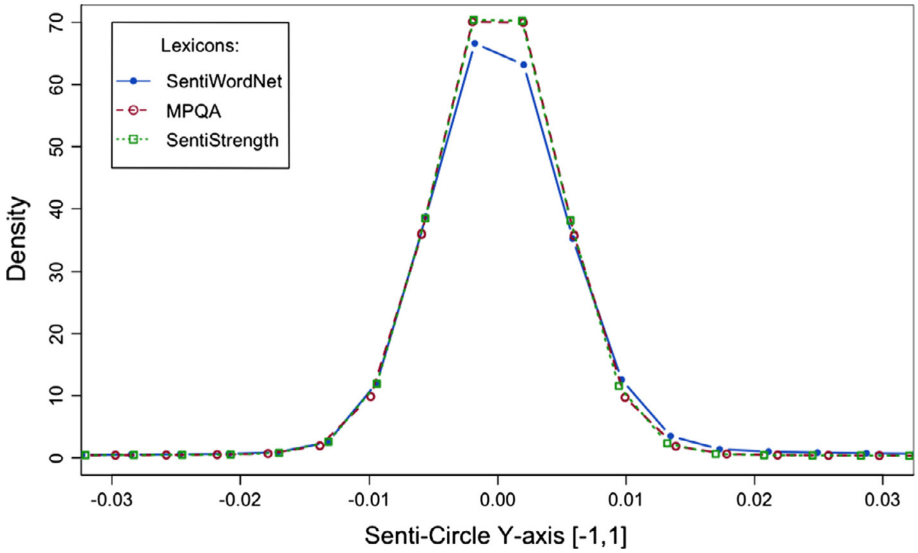


Fig. 5. Density geometric distribution of terms on the OMD dataset.

Table 4
Entity-level sentiment analysis results.

Methods	Accuracy	Subjective			Objective			Average		
		P	R	F1	P	R	F1	P	R	F1
Subjectivity classification (subjective vs. objective)										
MPQA-Method	63.79	67.27	92.50	77.89	0	0	0	33.64	46.25	38.95
SentiWordNet-Method	63.79	67.27	92.50	77.89	0	0	0	33.64	46.25	38.95
SentiStrength	62.07	64.15	91.89	75.56	40.00	9.52	15.38	52.08	50.71	51.38
Senti-Median (SentiWordNet)	**81.03**	90.91	78.95	84.51	68.00	85.00	75.56	79.45	81.97	**80.03**
Senti-Median (MPQA)	77.59	90.00	72.97	80.60	64.29	85.71	73.47	77.14	79.34	77.03
Senti-Median (Thelwall-Lexicon)	79.31	84.85	80.00	82.35	72.00	78.26	75.00	78.42	79.13	78.68
		Positive			Negative			Average		
Polarity classification (positive vs negative)										
MPQA-Method	72.5	80	92.31	85.71	71.43	45.45	55.56	75.71	68.88	70.63
SentiWordNet-Method	77.50	88.00	88.00	88.00	75.00	75.00	75.00	81.50	81.50	81.50
SentiStrength	85.00	95.65	81.48	88.00	70.59	92.31	80.00	83.12	86.89	84.00
Senti-Median (SentiWordNet)	**87.50**	89.29	92.59	90.91	83.33	76.92	80.00	86.31	84.76	**85.45**
Senti-Median (MPQA)	85.00	86.21	92.59	89.29	81.82	69.23	75.00	84.01	80.91	82.14
Senti-Median (Thelwall-Lexicon)	82.50	85.71	88.89	87.27	75.00	69.23	72.00	80.36	79.06	79.64

Bold values indicate highest accuracy and F-measure.

Table 5
Neutral region boundaries for *Y*-axis.

	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]

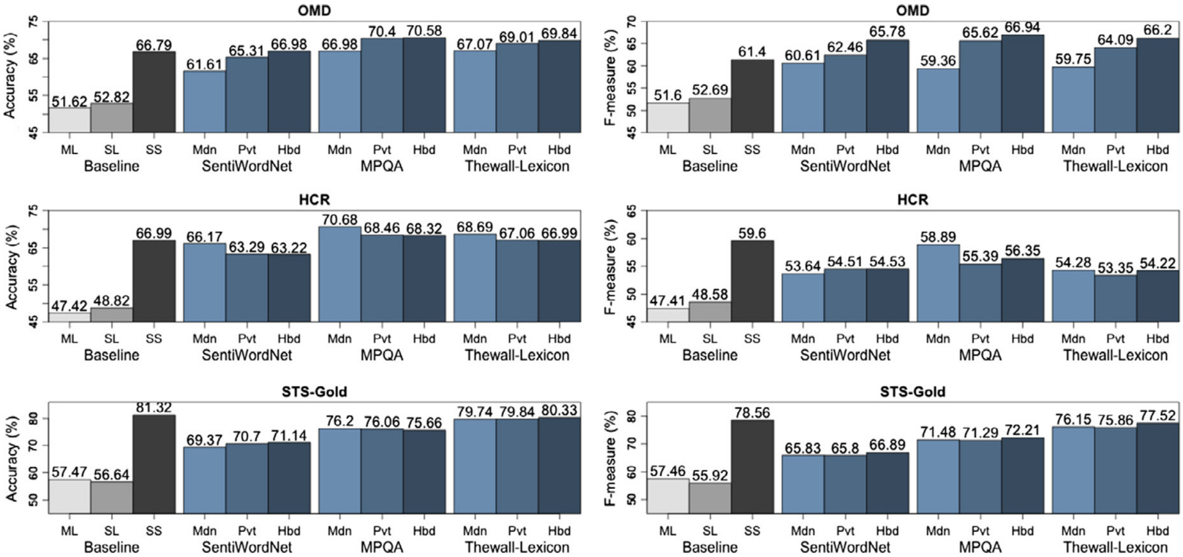


Fig. 6. Tweet-level sentiment detection results (accuracy and *F*-measure), where ML: MPQA-Method, SL: SentiWordNet-Method, SS: SentiStrength, Mdn: SentiCircle with Median method, Pvt: SentiCircle with Pivot method, Hbd: SentiCircle with Pivot-Hybrid.

baseline in both accuracy and average *F*-measure on all the datasets. On the OMD dataset, we observe a trend that Median < Pivot < Pivot-Hybrid. Both of our Pivot and Pivot-Hybrid methods give an average performance gain of 3.17% in accuracy and 4.3% in *F*-measure over SentiStrength with word prior sentiments obtained from either MPQA or Thelwall-Lexicon. The Median method does not bring any performance gain over SentiStrength. Overall, the best result is achieved by our Hybrid-Pivot method with word prior sentiments obtained from MPQA. It outperforms SentiStrength by nearly 5% in accuracy and 6% in *F*-measure.

On the HCR dataset all our three methods gave higher accuracy than SentiStrength, with word prior sentiments obtained from either MPQA or Thelwall-Lexicon. In particular, The Median method coupled with MPQA outperforms SentiStrength by nearly 4% in accuracy. In terms of *F*-measure, the Median method based on MPQA gives a similar result as SentiStrength.

While the best sentiment classification accuracy on the OMD or HCR datasets is only about 70%, we managed to achieve 80% on the STS-Gold dataset in sentiment classification accuracy. We observe that the Pivot-Hybrid method outperforms both the Median and the Pivot methods, regardless of where the word prior sentiments are obtained from, for the STS-Gold dataset. Nevertheless, the Pivot-Hybrid method using Thelwall-Lexicon gives 1% lower accuracy and *F*-measure than SentiStrength which uses the same lexicon.

The above results show a close competition between our three SentiCircle methods and the SentiStrength method. 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% and 66.56% respectively.

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 evaluation dataset contain more negative tweets than positive ones with the number of the former more than double the number of the latter (see Table 1).

6.3. Impact on words' prior sentiment

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 6 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.

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 6 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.

6.4. Runtime analysis

In this section we perform runtime analysis of the various methods and algorithms that constitute our SentiCircle model. To this end, we apply our model to the STS-Gold dataset using a computer with a i7 core CPU 2.3 GHz and 8G memory. According to the results reported in Table 7, building a term-index out of 2034 tweets (5035 unique terms) takes 13.8 s. This is not surprising given that this task involves several subtasks (tokenization, part-of-speech tagging, negation,

Table 6
Average percentage of words in three datasets, which their sentiment orientation or strength were updated by their SentiCircles.

	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 7
Runtime analysis of the SentiCircle model on the STS-Gold dataset.

Task	Run time
Term-index(generate)	13.8 s
Term-index(update)	6.6 ms
SentiCircle	47.18 ms
Senti-Median method	10 ms
Median method	16.9 ms
Pivot method	0.01 ms
Pivot-Hybrid method	4.53 ms

Non-English text filtering). We also construct context-term vectors and extract the terms' contextual features during this task as they all happen together. Although 13.8 s sounds relatively slow, building a term-index for a tweet collection is a one-time task, i.e., once it is built, the term-index can be used to perform various tasks in a relatively short time. For example, composing a SentiCircle from the term-index takes only 47.18 ms. Moreover, the term-index can be updated with new tweets on the fly, where it takes 6.6 ms to add a new tweet to the index.

We also estimate the runtime of our sentiment detection methods at entity and tweet levels. For entity-level, calculating Senti-Median takes 10 ms per entity. For tweet-level, the Median method takes 16.9 ms per tweet while the Pivot and Pivot-Hybrid methods take 0.01 and 4.53 ms respectively.

7. Discussion and future work

We showed the potential of using SentiCircles for sentiment detection at the entity and tweet levels. Due to the sheer lack of gold-standard datasets for evaluating entity-level sentiment, we constructed our own. We hope this will encourage further of such datasets to be made and released to help us expand this part of our evaluation, which is showing a high increase in performance in comparison to other methods.

We have seen that merely using MPQA or SentiWordNet for sentiment labelling fails to detect any neutral entities. SentiCircles, on the other hand, was able to amend sentiment scores of words in both lexicons based on contexts – hence achieving a much higher performance in detecting neutral entities than all the baselines. We plan to compare our approach against more sophisticated methods including machine learning ones, such as those based on Support Vector Machine (SVM) and Maximum Entropy (MaxEnt) classifiers. We also plan to evaluate our approach on new datasets of entities of different topic foci.

At the tweet-level, the evaluation was performed on three Twitter datasets and using three different sentiment lexicons. The results showed that our SentiCircle approach outperforms significantly the MPQA-Method and SentiWordNet-Method. Compared to SentiStrength, the results were not as conclusive, since SentiStrength slightly outperformed SentiCircles 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. The 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. As a future work, we plan to further investigate these issues and their influence on the performance of our approach.

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

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.

With regards to scalability, we plan to test our model on Twitter streams, which will require further optimisation steps such as controlling the size of the term-index by keeping only trending terms and removing fading ones, and updating the SentiCircles directly with new terms rather than re-computing them.

8. Conclusions

In this paper, we proposed a novel semantic sentiment representation of words, called SentiCircle, which is able to assign context-specific sentiment orientation to words. We described the use of SentiCircles for lexicon-based sentiment identification at both entity-level and tweet-level using different methods. Our proposed approach outperformed other lexicon labelling methods for both entity-level and tweet-level sentiment detection. For tweet-level sentiment detection, our approach also gave a better overall result than the state-of-the-art lexicon-based approach SentiStrength on two out of three datasets. While SentiStrength uses a fixed set of lexicon words and keeps the strength of each sentiment term unchanged across different data, our SentiCircle representation effectively updated the sentiment strength of many terms dynamically based on their contextual semantics in tweets.

Acknowledgments

This work was supported by the EU-FP7 project SENSE4US (Grant No. 611242).

YH is partially supported by the Shenzhen International Cooperation Research Funding (Grant No. GJHZ20120613110641217).

References

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of Twitter data. In *Proc. ACL 2011 workshop on languages in social media*. Portland, Oregon.

Andreevskaia, A., & Bergler, S. (2006). Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *Proceedings of EACL*. Trento, Italy.

Aue, A., & Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of recent advances in natural language processing (RANLP)*. Borovets, Bulgaria.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). 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.

Barbosa, L., & Feng, J. (2010). Robust sentiment detection on Twitter from biased and noisy data. In *Proceedings of COLING*. Beijing, China.

Batra, S., & Rao, D. (2010). Entity based sentiment analysis on Twitter. *Science*, 1–12.

Bifet, A., & Frank, E. (2010). Sentiment knowledge discovery in Twitter streaming data. In *Discovery science*. Canberra, Australia.

Cambria, E. (2013). An introduction to concept-level sentiment analysis. In *Advances in soft computing and its applications* (pp. 478–483). Springer.

Cambria, E., Havasi, C., & Hussain, A. (2012). Senticnet 2: A semantic and affective resource for opinion mining and sentiment analysis. In *FLAIRS conference* (pp. 202–207).

Diakopoulos, N., & Shamma, D. (2010). Characterizing debate performance via aggregated Twitter sentiment. *Proc. 28th int. conf. on human factors in computing systems*. ACM.

Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the international conference on Web search and web data mining*. Palo Alto, California, USA.

Firth, J. R. (1930-1955). A synopsis of linguistic theory. *Studies in Linguistic Analysis*, 1–32.

Garcia-Moya, L., Anaya-Sanchez, H., & Berlanga-Llavori, R. (2013). Retrieving product features and opinions from customer reviews. *IEEE Intelligent Systems, 28*(3), 19–27.

Go, A., Bhayani, R., & Huang, L. (2009). *Twitter sentiment classification using distant supervision*. CS224N project report, Stanford.

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). Target-dependent Twitter sentiment classification. In *Proceedings of the eighth conference on European chapter of the association for computational linguistics*. Portland, Oregon.

Kouloumpis, E., Wilson, T., & Moore, J. (2011). Twitter sentiment analysis: The good the bad and the omg! In *Proceedings of the ICWSM*. Barcelona, Spain.

Krippendorff, K. (2004). *Content analysis, an introduction to its methodology* (2nd ed., pp. 211–256). Thousand Oaks, CA: Sage Publications.

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of Natural Language Processing, 2*, 568.

Meng, X., Wei, F., Liu, X., Zhou, M., Li, S., & Wang, H. (2012). Entity-centric topic-oriented opinion summarization in Twitter. In *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining*. Beijing, China.

Neviarouskaya, A., Prendinger, H., & Ishizuka, M. (2011). Sentiful: A lexicon for sentiment analysis. *IEEE Transactions on Affective Computing*, 22–36.

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of LREC*. Valletta, Malta.

Pennebaker, J. W., Mehl, M. R., & Niederhoffer, K. G. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual Review of Psychology, 54*(1), 547–577.

Saif, H., He, Y., & Alani, H. (2012). Alleviating data sparsity for Twitter sentiment analysis. In *Proceedings, 2nd workshop on making sense of microposts (#MSM2012) in conjunction with WWW 2012*. Layon, France.

Saif, H., He, Y., & Alani, H. (2012). Semantic sentiment analysis of Twitter. In *Proceedings of the 11th international conference on the semantic web*. Boston, MA.

Saif, H., Fernandez, M., He, Y., & Alani, H. (2013). Evaluation datasets for Twitter sentiment analysis a survey and a new dataset, the sts-gold. In *Proceedings, 1st workshop on emotion and sentiment in social and expressive media (ESSEM) in conjunction with AI * IA conference*. Turin, Italy.

Saif, H., Fernandez, M., & Alani, H. (2014). Automatic stopword generation using contextual semantics for sentiment analysis of Twitter. In *The 13th semantic web conference (ISWC)*. Trentino, Italy.

Saif, H., He, Y., Fernandez, M., & Alani, H. (2014). Semantic patterns for sentiment analysis of Twitter. In *The 13th semantic web conference (ISWC)* (pp. 324–340).

Saif, H., Fernandez, M., He, Y., & Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of Twitter. In *Proc. 9th language resources and evaluation conference (LREC)*. Reykjavik, Iceland.

Saif, H., Fernandez, M., He, Y., & Alani, H. (2014). Senticircles for contextual and conceptual semantic sentiment analysis of Twitter. In *Proc. 11th extended semantic web conf. (ESWC)*. Crete, Greece.

Speriosu, M., Sudan, N., Upadhyay, S., & Baldridge, J. (2011). 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.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics, 37*(2), 267–307.

Takamura, H., Inui, T., & Okumura, M. (2005). Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd annual meeting on association for computational linguistics*.

Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology, 63*(1), 163–173.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology, 61*(12), 2544–2558.

Turney, P. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting of the association for computational linguistics (ACL'02)*. Philadelphia, Pennsylvania.

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

Turney, P. D., Pantel, P., et al. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37*(1), 141–188.

Wilson, T., Wiebe, J., & Hoffmann, P. (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*. Vancouver, British Columbia, Canada.

Wittgenstein, L. (1953). *Philosophical investigations*. London, UK: Blackwell (2001).

Xu, T., Peng, Q., & Cheng, Y. (2012). Identifying the semantic orientation of terms using S-HAL for sentiment analysis. *Knowledge-Based Systems, 35*, 279–289.