Enriching Lexicons with Ephemeral Words for Sentiment Analysis in Social Streams

Damianos P. Melidis
Faculty of Electrical Engineering and Computer Science & L3S Research Center, Leibniz University Hannover.
melidis@l3s.de

Alvaro Veizaga Campero
Faculty of Electrical Engineering and Computer Science & L3S Research Center, Leibniz University Hannover.
campero@l3s.de

Vasileios Iosifidis
Faculty of Electrical Engineering and Computer Science & L3S Research Center, Leibniz University Hannover.
iosifidis@l3s.de

Eirini Ntoutsi
Faculty of Electrical Engineering and Computer Science & L3S Research Center, Leibniz University Hannover.
ntoutsi@l3s.de

Myra Spiliopoulou
Faculty of Computer Science, Otto-von-Guericke-University Magdeburg.
myra@ovgu.de

ABSTRACT

Lexical approaches for sentiment analysis like SentiWordNet rely upon a fixed dictionary of words with fixed sentiment, i.e., sentiment that does not change. With the rise of Web 2.0 however, what we observe more and more often is that words that are not sentimental per se, are often associated with positive/negative feelings, for example, "refugees", "Trump", "iphone". Typically, those feelings are temporary as responses to external events; for example, "iphone" sentiment upon latest iphone version release or "Trump" sentiment after USA withdraw from Paris climate agreement.

In this work, we propose an approach for extracting and monitoring what we call *ephemeral words* from social streams; these are words that convey sentiment without being sentimental and their sentiment might change with time. Such sort of words cannot be part of a lexicon like SentiWordNet since their sentiment has an ephemeral character, however detecting such words and estimating their sentiment can significantly improve the performance of lexicon-based approaches, as our experiments show.

CCS CONCEPTS

• **Information systems** → **Social networks**; *Data stream mining*;

KEYWORDS

sentiment classification, dictionary-based approaches, ephemeral words, lexicon enrichment

ACM Reference Format:
Damianos P. Melidis, Alvaro Veizaga Campero, Vasileios Iosifidis, Eirini Ntoutsi, and Myra Spiliopoulou. 2018. Enriching Lexicons with Ephemeral Words for Sentiment Analysis in Social Streams. In *WIMS '18: 8th International Conference on Web Intelligence, Mining and Semantics, June 25–27, 2018, Novi Sad, Serbia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3227609.3227664

WIMS '18, June 25–27, 2018, Novi Sad, Serbia

ACM ISBN 978-1-4503-5489-9/18/06…$15.00
https://doi.org/10.1145/3227609.3227664

1 INTRODUCTION

Sentiment analysis aims at characterizing the sentiment content of a text as either positive or negative and is of paramount importance nowadays due to the huge amount of opinions that is generated on a daily basis from the social media like Twitter or Facebook [9]. Recognizing sentiment is a challenging task for machines due to implications like subjectivity of text and unambiguous phrasing.

Due to its importance and complexity, the domain has attracted a lot of research in the last years. The proposed approaches can be categorized into lexicon-based, machine learning and hybrid approaches. Methods in the first category use an existing, fixed and curated dictionary of words and their associated sentiment. The machine learning methods build a sentiment learning model from a given training set of labeled instances. Finally, the hybrid approaches comprise a combination of the two latter methods. Lexicon-based approaches have the advantage of relying upon qualitative resources however building and maintaining such a dictionary can be a tedious task. On the other hand, machine learning approaches are dynamic and depending on the training set they might also learn contextual features whereas lexicon-based approaches rely upon fixed-sentiment dictionaries.

In this paper we present a hybrid approach that combines a lexicon-based approach with a machine learning approach. We use SentiWordNet [1] as our sentiment lexicon. However, SentiWordNet models only a subset of the words in our data and more importantly, words with fixed sentiment. With the rise of Web 2.0 however, what we observe more and more often is that words that are not sentimental per se, are often associated with positive/negative sentiment, for example, "refugees", "Trump", "iphone". Our idea is to leverage such words for classification.

In particular, we first extract such words from a social stream, we refer to those words as *ephemeral words* since typically their sentiment is temporal or ephemeral as a response to external events. For example, sentiment of "Trump" fluctuates a lot following external events like election and travel ban. In order to predict the sentiment of a new document, we use two resources: SentiWordNet for words

of fixed sentiment and the pool of ephemeral words. For the latest, we build for each word, an ephemeral word trajectory that models the sentiment history of the word and can be used to make a confident estimation about word's current sentiment. In other words, we combine words with fixed/static sentiment (from SentiWordNet) with words with dynamic/ephemeral sentiment (from the pool of ephemeral words).

The rest of the paper is organized as follows: Related work is discussed in Section 2. Our approach is presented in Section 3. Experimental results are shown in Section 4. Conclusion are discussed in Section 5.

2 RELATED WORK

Methods for sentiment learning from tweets or, short snippets of text in general, can be divided into three categories: lexicon-based approaches, machine learning approaches and hybrid approaches. In the following, we will briefly discuss these categories.

2.1 Lexicon-based approaches

Lexicon-based approaches match the words of the document to words of a manually curated sentiment dictionary and aggregate their associated sentiment to predict the total sentiment of the document. A well-known sentiment specific dictionary is *SentiWordNet* [1]. The SentiWordNet sentiment dictionary extends the WordNet lexical database by associating each word per its meaning to a negative, positive and neutral sentiment. Many approaches of this category have used the SentiWordNet dictionary. For example, researchers in [15] acquired the sentiment for each word of the document from SentiWordNet then they used a rules-based system to combine the sentiment and predict the overall sentiment of the document/tweet.

Because those dictionaries are curated their information is very qualitative. However their coverage is not that high and moreover they cannot represent contextual information. Typically in these approaches, the sentiment orientation of a word is independent from its context. Moreover, the sentiment of the words is fixed.

Several approaches have been proposed to overcome the aforementioned limitations. For example, lexical databases are used to expand the lexicons using the semantic relations that exist among the words [18]. Mainly, lexicons are expanded under the assumption that words that are synonyms have the same sentiment whereas words that are antonyms have the opposite. Online dictionaries like WordNet are commonly used for such an expansion. Other approaches expand lexicons using statistical approaches based on the information available in the corpus. The intuition behind this approach is that words that occur multiple times in documents of the same label are likely to have the same sentiment orientation as the label. Point Wise mutual information (PMI) is commonly used by such approaches, for example inf [11]. The semantic polarity of a word is defined by the difference between the PMI of all the other words which are in the neighborhood of the given word and can be found in a sentiment dictionary.

2.2 Machine Learning approaches

Machine learning approaches for sentiment analysis [16] build sentiment learning models from a training dataset and apply those models upon new documents of unknown sentiment in order to predict their sentiment. Depending on the availability of training labels, such methods can be further divided into (fully) supervised and semi-supervised learning methods. The majority of the work is on supervised learning, where the different methods differ mainly w.r.t the employed features and learning models. For example, [16] investigates different text representations and three learning algorithms (Naive Bayes, Maximum Entropy and Support Vector Machines). Recently, deep learning methods are often employed, for example, in [20] the authors encode the sentiment information into the representation of the word embeddings and then incorporate this representation to a convolutional neural network (CNN).

2.3 Hybrid approaches

Hybrid approaches comprise a combination of lexicon-based and machine learning approaches. For instance, in [4, 14] the authors use the strength polarity of terms found in lexicons as additional features for training a classifier. Also, the best performing algorithm for SemEval 2013 [13] created additional lexicons, one from hashtagged tweets and a second one from tweets with emoticons, used *n*-grams representation for both words and letters and exploited additional syntactical features of the tweets like capital letters and punctuation. In [10] the authors propose an approach for sentiment analysis in Twitter that leverages entity information. In particular, the authors compute the sentiment for entities such as "Obama" and "iPad" based on the proximity of words with known sentiment from a sentiment dictionary. Our approach also falls into this category.

2.4 Streaming approaches

The majority of sentiment analysis approaches, especially the machine learning based and the hybrid ones, refer to the static or batch case based on the assumption all data is available in advance. Except for the batch/static approaches, there also exist incremental approaches for sentiment analysis over textual streams. For example [2] deployed different learners, namely Hoeffding trees, Stochastic Gradient Descent (SGD) and Multinomial Naive Bayes (MNB) to predict the sentiment of tweets coming from the Twitter stream. A known challenge for streams is the occurrence of concept drifts, i.e., changes in the underlying data population that affect the learning models. Stream classifiers can adapt to such changes [19] either implicitly, the so-called blind adaptation methods , e.g., [22] or explicitly, the so-called informed adaptation methods, e.g., [8]. The former update the model constantly based on new instances, while the later update the model only when concept drift is detected.

Our work lies in the category of *hybrid* approaches. We identify those sentimental words that are not part of a sentiment lexicon like SentiWordNet, the so-called *ephemeral* words. For each word, we keep track of its sentiment over time, modeled as a timeseries. Our motivation lies in the fact that these words might have *ephemeral* sentiment scores. Consequently, we learn the sentiment of such *ephemeral* words using simple statistics over their time series. Finally, we combine the predicted sentiment of such words with the sentiment of the remaining words, of the tweet, which can be found in a sentiment dictionary like SentiWordNet.

3 EXPLOITING WORD HISTORIES FOR SENTIMENT ANALYSIS IN SOCIAL STREAMS

An overview of our approach is shown in Figure 1. As shown in this figure, we process the social stream each tweet arriving at a distinct time point t_i. As several tweets may arrive at the same time point we process the tweets into batches, $D_1, \cdots, D_k, \cdots$ arriving at time points $t_1, \cdots, t_k, \cdots$, respectively. Each batch D_k consists of documents $d_i \in D_k$ with sentiment class label $c(d_i) \in \{pos, neg\}$. At this scale, the class labels are typically weak labels or proxies to the real class labels. What is commonly used especially in Twitter streams is emoticons as proxies for the class labels, e.g., [6].

Let V_k be the vocabulary of distinct words of batch D_k; not all words in V_k are part of SentiWordNet. For the words in the *difference set*, our idea is to use a machine learning approach to estimate their sentiment from the stream (Section 3.1). For those words, we keep track of their sentiment at each batch (Section 3.2) and generate what we call word histories (Section 3.3), i.e., a time series of the word sentiment over time. We use those histories, to predict the

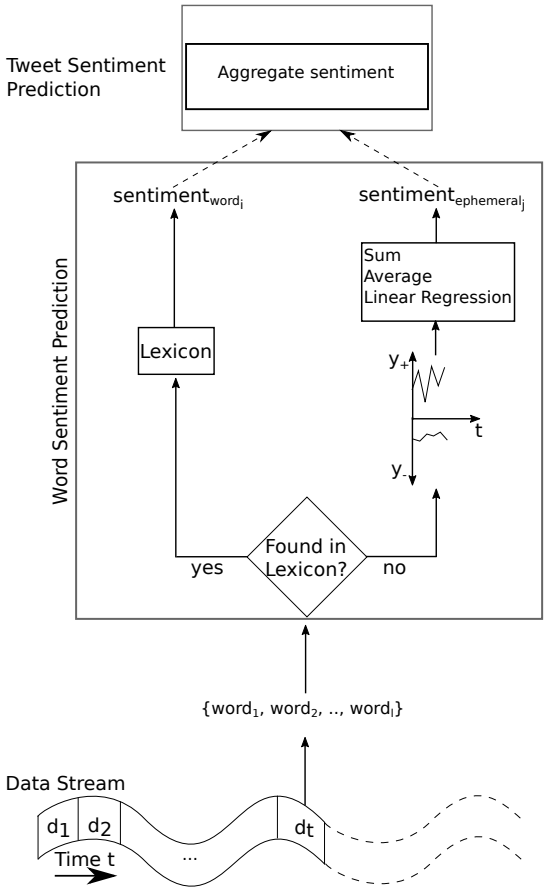

Figure 1: Overview of our approach.

sentiment of words not in SentiWordNet. When a new document arrives, we use SentiWordNet estimations for words already in SentiWordNet, whereas for words not in SentiWordNet, we predict their sentiment based on word histories (Section 3.4). Therefore, our approach enriches lexicon-based approaches with temporal predictions of ephemeral words.

3.1 Extracting ephemeral sentiment words

Not all the words in documents of batch D_k are part of SentiWordNet; words like "Trump" or "iphone" are not sentimental per se, but they are associated with positive/negative feelings over certain periods of time.

All those words from the batch D_k that are not contained in SentiWordNet, are potential words of interest for sentiment tracking, we denote them by E_k. We refer to these words as *ephemeral words*, where the term *ephemeral* refers to the temporal trend of their sentiment.

3.2 Estimating the sentiment of ephemeral sentiment words at each time point

For each word $e \in E_k$ we estimate its sentiment at a time point t_k using the sentiment of the documents containing e in D_k. Let $freq(e, c)$ be the frequency of word e in documents of class c in D_k, $c \in \{pos, neg\}$.

The probability of a word e being in class c, is estimated by the percentage of occurrences of e in documents of class c:

$$p(e)_c = \frac{freq(e, c)}{\sum_{c \in \{pos, neg\}} freq(e, c)} \quad (1)$$

In other words, we normalize the occurrences of word e in class c with the overall occurrences of word e in batch D_k. The result is two estimations, one for the positive and one for the negative class.

Thus far, the frequency term of Equation 1 refers to word frequency (summing up the word occurrences in each document). We employ another version too that counts the number of tweets containing a word rather than the number of occurrences of that word in the tweets. We refer to the first as *token-based* and to the second as *tweets-based* version. In our experiments, there was no big difference between the two versions. This is probably due to the 140 characters limit in Twitter, consequently users typically avoid repetitions within a tweet for space sake.

3.3 Ephemeral sentiment word trajectories

For each extracted ephemeral word e, we estimate its sentiment values at each time point and synthesize what we call the sentiment trajectory of an ephemeral word or simply, *ephemeral word trajectory*.

The time points correspond to word observations in the stream, whereas the time series values is the sentiment associated with these time points, estimated as described above. For an ephemeral word e with observation time points $\{s_1, s_2, \cdots, s_t\}$ up to current time point s_t, the corresponding sentiment trajectory will be:

$$trajectory(e) = \{v_1, v_2, \cdots, v_t\}$$

where v_i is the sentiment vector of the word e at time point t_i. In order to capture different periodicities of the trajectories we applied a sliding window, h, of the past observations to predict the current value. For example, given the current time point v_t we kept

a sliding window of 1 day, $|h| = 1$ day. That is:

$$\text{sliding trajectory}(e) = \{v_i, v_{i+1}, \cdots, v_{t-1}\}, \text{ where i} : s_{t-1} - s_i \leq |h|.$$

For the v_{t+1} we slided the window again in order to accomplish that the difference between the last and the first time point of the trajectory is lower or equal to $|h|$.

Note that there might be gaps in the monitoring period, as a word e might not be observed at all time points from the stream.

3.4 Leveraging ephemeral sentiment words for sentiment classification

The ephemeral word trajectories can be used as a standalone tool for understanding the sentiment evolution of a given word e and for predicting its sentiment at a next time point. Moreover, those predictions can be incorporated into the sentiment classification process of dictionary-based approaches: As already mentioned, SentiWordNet is a fixed lexicon. One can leverage the pool of ephemeral words in order to consider for the classification, words beyond those covered by SentiWordNet. That is, in order to classify a new tweet d, we take into account both:

- words $w \in d$ which occur in SentiWordNet. For those, we use their SentiWordNet sentiment values.
- words $w \in d$ which do not occur in SentiWordNet but are tracked as ephemeral words. For those, we estimate their sentiment value at the given time based on their sentiment trajectory.

For the later, we applied different prediction models, namely, simple average and linear regression over the past sentiment values, $v_1, \cdots, v_{t-1}$, in order to predict the current sentiment value, v_t, of the word.

One could apply more complex prediction models, e.g., by capitalizing upon the vast amount of work on time series like [12]; we leave it as part of our future work.

3.5 Implementation

For the identification and extraction of ephemeral words we used `Scala`. Independently, for each extracted ephemeral word we learned from its trajectory using `Java`. The final step of classification of each tweet was performed using `Scala`. All `Scala` code is available at a public github repository[1].

4 EXPERIMENTS

4.1 Datasets & Preprocessing

For our experiments we used three datasets, a small human-labeled dataset and two bigger machine-labeled ones. The details of the datasets follow, as well as the preprocessing steps we undertook.

The Stanford Twitter Sentiment STS dataset [6]: was created by querying the Twitter API for messages between April 6, 2009 and June 25, 2009 introduced in [6]. The sentiment labels were derived via distant supervision by using emoticons as proxies for labels [6]. In particular, tweets containing positive emoticons like ":)" were marked as positive and tweets containing negative emoticons like ":(" were marked as negative. Tweets that did not have any of these labels or had both, were discarded. The final stream consists of 1.6M opinionated tweets, 50% of which are positive and 50% negative.

The STS-Gold dataset [17]: was created by randomly selecting 2,034 tweets from the STS dataset [6]. For each tweet, its corresponding words were extracted using the AlchemyAPI[2]. The 2,034 tweets were manually labeled at both the tweet-level and at the word-level with respect to their sentiment. The STS-Gold dataset is small and has no temporal information but it contains qualitative human labels.

The BigTwitter dataset: We collected data from Twitter using its public streaming API[3], which provides a random selection of tweets (about 1% of all tweets). In total we have 45 months of tweets in our dataset (from 2013-04-01 until 2016-12-31). We focus on English tweets. The dataset is automatically labeled via co-training as described in [7]. The total volume of the dataset is 1,327,139,507 tweets. We use this dataset to investigate the characteristics of ephemeral words (Section 4.2) rather than for evaluating the performance of our hybrid approach. For the later and for efficiency reasons, we use a filtered version of the dataset described hereafter.

The Twitter 2015 focused dataset: The filtered dataset is extracted from the BigTwitter dataset for a period of 12 months (from 2015-01 till 2015-12) and based on tweets that contain the following predefined words/entities: "Obama", "Merkel", "Tsipras", "CNN", "Messi", "blacklives-matter", "isis" and "syrianrefugees". The total volume of the dataset is 931,498 tweets. The dataset is imbalanced with 92.4% positive vs 7.6% negative tweets.

To improve the quality of our data, we applied several preprocessing steps like stopword removal, low frequency pruning etc, similarly to [7]:

- *Mentions, urls, hashtags and html chars removal:* We removed all mentions (i.e., terms starting with "@") and urls. Also we treated hashtags (i.e., terms starting with "#") as normal words by removing the special char "#". In addition we removed all html tags.
- *Prefix filtering:* Words like "remake" and "predefine" contain prefixes. We filtered the words which contain such sort of prefixes and we kept only the ones in Roget's thesaurus[4].
- *Arithmetic chars removal:* We removed numbers and words containing numbers, except for words ending with numbers as many such words are associated with different product versions like "iphone6", "playstation3" or "android5" and reoccurring events like "xmas17", "xmas16".
- *Repeated chars:* We replaced repeated characters occurring more than two times by only one char, for example, "hiiiiiiii" was replaced by "hi".
- *Special chars removal:* Special characters such as "*","$", "!", etc were removed.
- *Stop words:* We filtered out common words using a predefined list of stop words[5].
- *Low frequency words:* Words that occurred less than 10 times (≤ 10) were omitted.
- *Short tweets:* We omitted tweets with less than 4 words based on [21].

[1] https://github.com/damianosmel/ExtractingEphemeralEntities.git

[2] www.alchemyapi.com/

[3] https://dev.twitter.com/streaming/overview

[4] http://www.roget.org/

[5] http://weka.sourceforge.net/doc/dev/weka/core/stopwords/Rainbow.html

The effect of preprocessing is shown in Figure 2, only for the BigTwitter dataset. As we can see, removing stop words, low frequency words and very short tweets has a strong effect on the number of words.

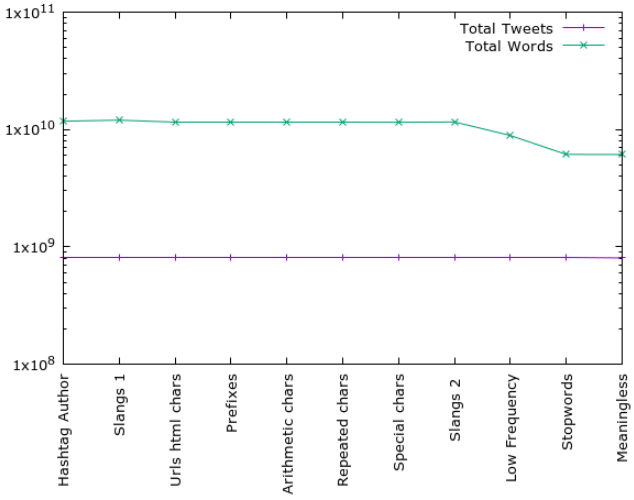

Figure 2: BigTwitter dataset: Effect of preprocessing.

For the STS-Gold dataset, we report on both token-based and tweets-based approaches for estimating the sentiment value of a tracked word (Section 3.2). The results show that there is no big difference in the performance, therefore for the remaining datasets we report only on the token-based approach.

4.2 Ephemeral words extraction and tracking

For ephemeral word extraction and tracking we report on the BigTwitter dataset which spans a period of around three years. Instead of reporting on all words, we report only on named entities; these are words that correpond to real-world objects, such as persons, locations, organizations, products, etc. To identify such words we use the FEL entity annotator [3].

The amount of extracted ephemeral entities per month is shown in Figure 3, together with the total amount of words and the amount of words not in SentiWordNet. We observe that the SentiWordNet covers only a small fraction of all words (46.71 % on average). From this set, the extracted entities are only the 37.31%.

Not all entities are tracked over the whole observation period. To understand the longevity of the entities, we display the frequency distribution in Figure 4. The majority of the entities occurs just once (in one month) or during the whole observation period (45 months). In total, 89,455 ephemeral entities were extracted over the whole stream and about 14,000, 15.6%, ephemeral entities were present in the whole monitoring period of 45 months. Among these entities belong: persons like "bieber" and "obama", organizations/companies like "facebook" and "mcdonalds", products like "iphone" and "sony" and common words like "hey" and "ugh". In Table 1 we display an example of popular ephemeral entities tracked over the whole observation period.

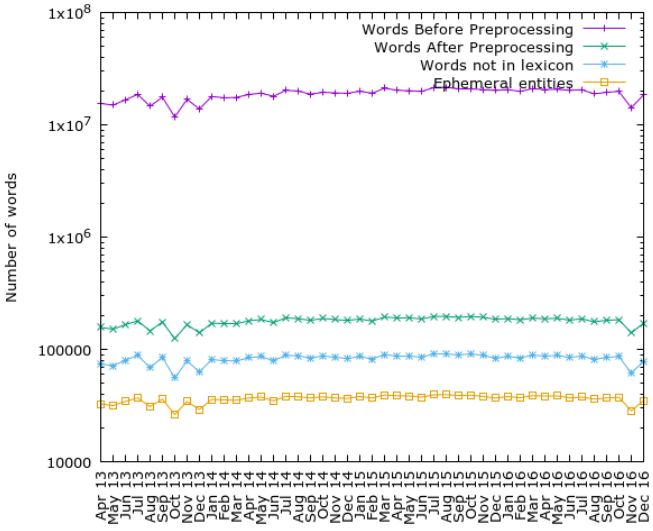

Figure 3: BigTwitter dataset: (Distinct) words, words not in SentiWordNet and ephemeral entities per month.

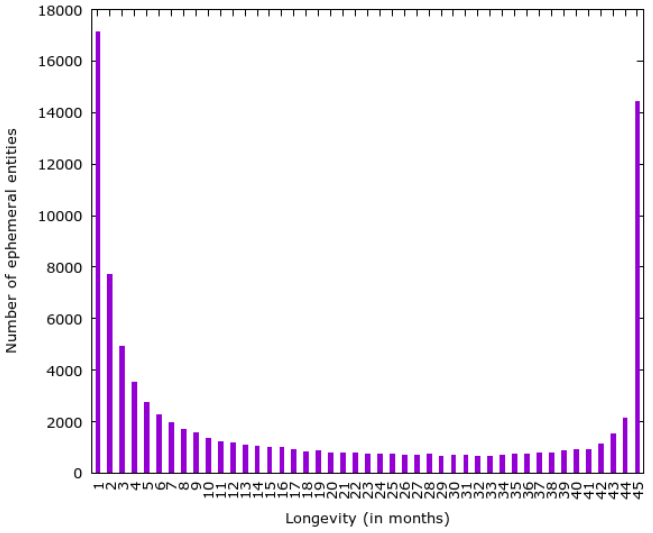

Figure 4: BigTwitter dataset: Longevity distribution of ephemeral entities.

To understand how the pool of ephemeral entities evolves over the stream, in Figure 5, we compare the entities extracted at each month to the entities extracted from the first month of the monitoring period (April 2013). The growth is computed as $\frac{|E_1 \cup E_i|}{|E_1|}$ for $2 \leq i \leq 45$ where E_1 is the set of ephemeral entities extracted from April 2013 and E_i is the ephemeral entity set extracted from month M_i. We observe that new ephemeral entities are tracked over time, which were not part of the ephemeral entities extracted from the first month (E_1). The smallest growth is observed in October 2013 (0.05%) and the largest in August 2015 (0.37%). We see that in 2013 the growth is smaller comparing to 2014-2016. This is probably

Table 1: BigTwitter dataset: An example of popular extracted ephemeral entities

ahh	michelle	absolut	tryin
hey	obama	wont	accenture
barack	kanye	vintage8	accessorize
bbc	facebook	rihanna	everytime
bieber	selena	starbucks	messi
gangnam	mcdonalds	superbowl	badass
merkel	microsoft	ugh	cutie
iphone	iran	sony	minecraft
kardashian	katy	trippin	lovin

due to the dynamic nature of Twitter; new entities are discussed over time also in response to external events. This observation strengthen our intuition that such sort of ephemeral entities cannot be part of a lexicon like SentiWordNet as they (most of them) probably have an "expiration" date and therefore they should be maintained separately in an online fashion. In Figure 6, we show the growth of the ephemeral entities pool between consecutive months M_i, M_{i+1}, $1 \leq i \leq 45 - 1$ computed as: $\frac{|E_i \cup E_{i+1}|}{|E_i|}$. There is a lot of fluctuation in 2013, in the rest of the observation period the growth seems more stable (on average 20% monthly growth).

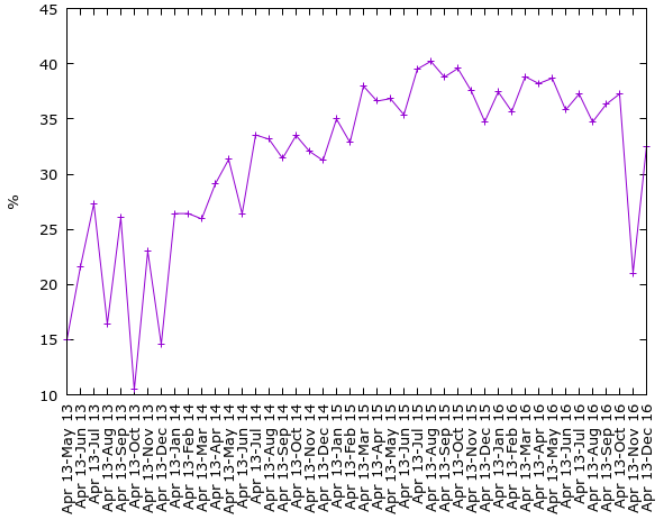

Figure 5: BigTwitter dataset: Ephemeral entities pool growth (w.r.t. first month).

4.3 The effect of ephemeral words on sentiment classification

As already mentioned, we could leverage the pool of ephemeral words to improve the performance of sentiment classification in dictionary-based approaches. We report on the results on the different datasets.

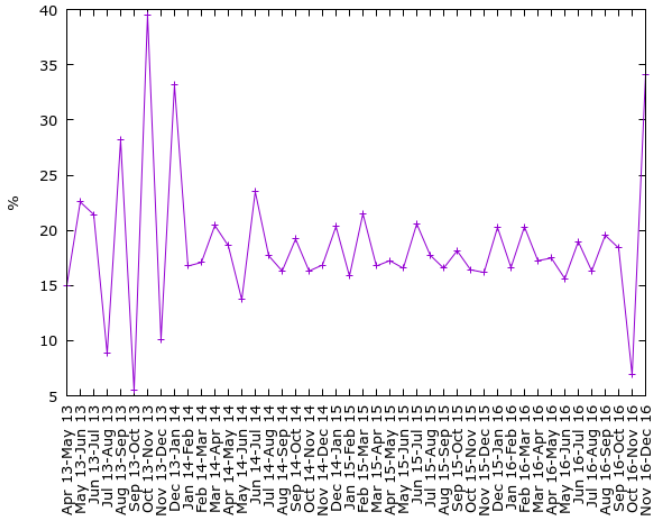

Figure 6: BigTwitter dataset: Ephemeral entities pool growth (w.r.t. previous month)

4.3.1 Effect on sentiment classification: STS-Gold dataset. The STS-Gold dataset is pretty small (2,034 tweets) and with no temporal information, therefore we extract the entities, words not in SentiWordNet which are also identified as named-entities, over the whole dataset instead using a sliding window of the time series of the word. As entity annotator for this case we used AlchemyAPI. The extracted entities were 18 compromising the 7.5% of all words of the data set after preprocessing. We estimated the sentiment of the aforementioned words using Equation 1 in its two variations (tokens-based and tweets-based). We compared the performance of classification based on SentiWordNet versus classification based on SentiWordNet and extracted entities, the results are shown in Table 2.

We can see that all proposed methods outperform the SentiWordNet baseline, in terms of overall accuracy and F1 scores. The tweets based version has 1.5 % higher F1 score compared to the base line. This approach is significantly different than the baseline with $p < 0.001$ based on Mc-Nemar's test.

Although the dataset is quite small, this result indicates that using the ephemeral words has a strong positive impact on sentiment classification.

4.3.2 Effect on sentiment classification: STS dataset. We process the STS dataset at different levels of granularity: daily, weekly and monthly. The results are shown in Table 3. In total, 15,251 words were extracted, from this amount only 1,246 words were used as input in the trajectories. We extract the ephemeral word trajectories for each granularity. We used the trajectory data to predict the sentiment of a word using sliding window of different sizes. That is to predict the sentiment value for the next time point t_{i+1} we used simple statistics average and linear regression over the sliding window of size $|h|$, where $|h|$ =1 day or 1 week or 1 month or 3 months.

Method	Accuracy	Precision	Recall	F1
SentiWordNet (baseline)	0.577	**0.870**	0.424	0.570
SentiWordNet & Ephemeral Words (tokens)	**0.662**	0.718	**0.487**	0.580
SentiWordNet & Ephemeral Words (tweets)	0.659	0.709	0.484	**0.585**

Table 2: STS-Gold dataset: Evaluation results.

Method	Accuracy	Precision	Recall	F1
SentiwordNet (Baseline)	0.627	0.782	0.590	0.673
SentiwordNet + Ephemeral Words(Avg 1 day)	0.635	0.837	0.592	0.693
SentiwordNet + Ephemeral Words(Regression 1 day)	**0.645**	0.847	**0.598**	0.701
SentiwordNet + Ephemeral Words(Avg 1 week)	0.636	0.846	0.591	0.696
SentiwordNet + Ephemeral Words(Regression 1 week)	0.644	0.855	0.597	**0.703**
SentiwordNet + Ephemeral Words(Avg 1 month)	0.638	0.856	0.591	0.699
SentiwordNet + Ephemeral Words(Regression 1 month)	0.641	**0.858**	0.594	0.702
SentiwordNet + Ephemeral Words(Avg 3 months)	0.638	0.857	0.591	0.700
SentiwordNet + Ephemeral Words(Regression 3 months)	0.641	0.858	0.594	0.702

Table 3: STS dataset: Evaluation results.

As we can see, all methods beat the SentiWordNet baseline with a regression model of 1 week giving the best F1 score (the best accuracy was achieved by a regression model of 1 day). This outperforming model is significantly different compared to the baseline with $p < 0.001$ based on Mc-Nemar's test.

Although the improvement is not drastic, we have to take into account that the set of tracked words is a small subset of the whole feature space (words) used for prediction. More specifically, the number of the tracked words 1,246 which is only the 5.1 % of the whole data set. Also the number of tweets with words tracked as ephemeral words is 68.34%. As the reported results are over the whole dataset, not only based on tweets with ephemeral words, we may state that these results are pessimistic for our method. In the following section we have shown this claim by using a data set which mostly contains tweets with tracked words.

4.3.3 Effect on sentiment classification: The Twitter 2015 focused dataset. We process the dataset using again different size of sliding window namely 1 day, 1 week, 1 month and 3 months. The results are shown in Table 4.

We extracted the ephemeral word trajectories using again sliding windows of the same sizes (one day or one week or one month or three months). We predict the sentiment value for the next time point t_{i+1} using the same methods (average and linear regression) over a sliding window.

It is noticeable that all methods beat the SentiWordNet baseline with the model of the average method of sliding window of size 3 months giving the best F1 score and accuracy. This outperforming model is significantly different compared to the baseline with $p < 0.001$ based on Mc-Nemar's test.

The improvement is more drastic in this data set, 15% improvement on accuracy, as the number of tracked (unique) words is 1,633 which corresponds to 7.84% of the whole feature space and 99.6% from the tweets in the dataset contain at least one tracked word. So, these results strengthen what we have observed with the other two datasets, i.e., that our hybrid approach performs better.

5 CONCLUSIONS AND OUTLOOK

We propose a hybrid approach for sentiment analysis in social streams which combines a lexicon of fixed sentiment words like SentiWordNet with a machine learning approach that learns the sentiment of words not in dictionary based on their history of observations in the different classes over the course of the stream. In particular, we leverage the ephemeral words (which represent temporal sentiment words) for sentiment classification together with lexicon words (which represent fixed-sentiment words) to improve sentiment classification of short documents in social streams.

Our results over three different datasets from Twitter show that our method outperforms the lexicon-based baseline. The improvement is different over the different datasets and it definitely depends on the amount of emphemeral words occurring in the new incoming documents from the stream. In more details, on the first dataset (STS-Gold) tracking and predicting for only the ephemeral entities, 7.5% of the whole number of words, our approach improved on 1.5% on F1 measure over the baseline. For the STS dataset tracking and predicting the ephemeral words which are only the 5.1% of the whole number of words, resulted to 3% hihger F1-score compared to the baseline. For the last dataset, Twitter 2015 focused, tracking and predicting ephemeral words, 7.84% of the whole feature space, improved the F1-score by 12% compared to the baseline.

In this work we do not elaborate upon the best way of using the trajectory for estimating the current sentiment of an ephemeral word. As future work, we will focus on methods to better estimate the sentiment over the trajectory of an *ephemeral* word. For example

Method	Accuracy	Precision	Recall	F1
SentiwordNet (Baseline)	0.502	0.532	0.883	0.664
SentiwordNet + Ephemeral Words (Avg 1 day)	0.593	0.632	0.897	0.742
SentiwordNet + Ephemeral Words (Regression 1day)	0.575	0.611	0.897	0.727
SentiwordNet + Ephemeral Words (Avg 1 week)	0.618	0.661	0.899	0.762
SentiwordNet + Ephemeral Words (Regression 1 week)	0.594	0.633	0.898	0.743
SentiwordNet + Ephemeral Words (Avg 1 month)	0.636	0.681	0.901	0.776
SentiwordNet + Ephemeral Words (Regression 1 month)	0.613	0.655	0.900	0.758
SentiwordNet + Ephemeral Words (Avg 3 months)	**0.650**	**0.697**	**0.902**	**0.786**
SentiwordNet + Ephemeral Words (Regression 3 months)	0.626	0.670	0.900	0.768

Table 4: The Twitter 2015 focused dataset: Evaluation results

instead of our simple methods for time series prediction we can use more complex models such as LSTMs [5].

6 ACKNOWLEDGEMENTS

The work was funded by the German Research Foundation (DFG) project OSCAR (Opinion Stream Classification with Ensembles and Active leaRners) and by the European Commission for the ERC Advanced Grant ALEXANDRIA under grant No. 339233.

REFERENCES

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.. In *LREC*, Vol. 10. 2200–2204.

[2] Albert Bifet and Eibe Frank. 2010. Sentiment knowledge discovery in twitter streaming data. In *International conference on discovery science*. Springer, 1–15.

[3] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. 2015. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 179–188.

[4] Yan Dang, Yulei Zhang, and Hsinchun Chen. 2010. A lexicon-enhanced method for sentiment classification: An experiment on online product reviews. *IEEE Intelligent Systems* 25, 4 (2010), 46–53.

[5] John Cristian Borges Gamboa. 2017. Deep Learning for Time-Series Analysis. *arXiv preprint arXiv:1701.01887* (2017).

[6] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1 (2009), 12.

[7] Vasileios Iosifidis and Eirini Ntoutsi. 2017. Large Scale Sentiment Learning with Limited Labels. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1823–1832.

[8] Vasileios Iosifidis, Annina Oelschlager, and Eirini Ntoutsi. 2017. Sentiment Classification over Opinionated Data Streams Through Informed Model Adaptation. In *International Conference on Theory and Practice of Digital Libraries*. Springer, 369–381.

[9] Jussi Karlgren, Magnus Sahlgren, Fredrik Olsson, Fredrik Espinoza, and Ola Hamfors. 2012. Usefulness of sentiment analysis. In *European Conference on Information Retrieval*. Springer, 426–435.

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

[11] Farhan Hassan Khan, Usman Qamar, and Saba Bashir. 2016. SentiMI: Introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection. *Applied Soft Computing* 39 (2016), 140–153.

[12] Atsutoshi Kumagai and Tomoharu Iwata. 2016. Learning future classifiers without additional data. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[13] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242* (2013).

[14] Bruno Ohana and Brendan Tierney. 2009. Sentiment classification of reviews using SentiWordNet. In *9th. IT & T Conference*. 13.

[15] Reynier Ortega, Adrian Fonseca, and Andres Montoyo. 2013. SSA-UO: unsupervised Twitter sentiment analysis. In *Second joint conference on lexical and computational semantics (SEM)*, Vol. 2. 501–507.

[16] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 79–86.

[17] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. 2013. Evaluation Datasets for Twitter Sentiment Analysis A survey and a new dataset, the STS-Gold. (2013).

[18] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. 2014. Senticircles for contextual and conceptual semantic sentiment analysis of twitter. In *European Semantic Web Conference*. Springer, 83–98.

[19] Myra Spiliopoulou, Eirini Ntoutsi, and Max Zimmermann. 2016. *Opinion Stream Mining*. Springer US, Boston, MA, 1–10. https://doi.org/10.1007/978-1-4899-7502-7_905-1

[20] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification.. In *ACL (1)*. 1555–1565.

[21] Pablo A Tapia and Juan D Velásquez. 2014. Twitter sentiment polarity analysis: A novel approach for improving the automated labeling in a text corpora. In *International Conference on Active Media Technology*. Springer, 274–285.

[22] Sebastian Wagner, Max Zimmermann, Eirini Ntoutsi, and Myra Spiliopoulou. 2015. Ageing-based multinomial naive bayes classifiers over opinionated data streams. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 401–416.