Transferring Sentiment Knowledge between Words and Tweets

Felipe Bravo-Marquez [*], Eibe Frank , and Bernhard Pfahringer
Department of Computer Science, University of Waikato, Hamilton, New Zealand
E-mail: fbravoma@waikato.ac.nz

Abstract. Message-level and word-level polarity classification are two popular tasks in Twitter sentiment analysis. They have been commonly addressed by training supervised models from labelled data. The main limitation of these models is the high cost of data annotation. Transferring existing labels from a related problem domain is one possible solution for this problem. In this paper, we study how to transfer sentiment labels from the word domain to the tweet domain and vice versa by making their corresponding instances compatible. We model instances of these two domains as the aggregation of instances from the other (i.e., tweets are treated as collections of the words they contain and words are treated as collections of the tweets in which they occur) and perform aggregation by averaging the corresponding constituents. We study two different setups for averaging tweet and word vectors: 1) representing tweets by standard NLP features such as unigrams and part-of-speech tags and words by averaging the vectors of the tweets in which they occur, and 2) representing words using skip-gram embeddings and tweets as the average embedding vector of their words. A consequence of our approach is that instances of both domains reside in the same feature space. Thus, a sentiment classifier trained on labelled data from one domain can be used to classify instances from the other one. We evaluate this approach in two transfer learning tasks: 1) sentiment classification of tweets by applying a word-level sentiment classifier, and 2) induction of a polarity lexicon by applying a tweet-level polarity classifier. Our results show that the proposed model can successfully classify words and tweets after transfer.

Keywords: Sentiment classification, Polarity lexicon expansion, Twitter, Transfer learning

1. Introduction

Twitter[1] is a widely-used microblogging service in which users post short messages, or tweets, limited to 140 characters[2] to express their opinions and thoughts. Automatic analysis of sentiment in tweets has potential applications in a wide range of fields such as business, sports, and politics. However, the brevity of tweets and the range of informal expressions frequently used in them, including slang words, hashtags, and emoticons, (e.g., lol, omg, hahaha, #hatemonday) make sentiment analysis of tweets a difficult task.

There are two sentiment analysis tasks for tweets that have received substantial attention:

1. **Message-level polarity classification** (MPC) [1], which is the task of classifying tweets into sentiment categories such as positive and negative. For example classifying tweets *"got paper accepted yey :)"* and *"This is horrible!!"* to the positive and negative class respectively.
2. **Polarity lexicon induction** (PLI) [2]: a polarity lexicon is a list of words labelled by sentiment. The PLI task consists of classifying words from a corpus of tweets into sentiment categories. For example, classifying words such as *lovee* and *hapyyy* to the positive class, and words like *#hateyou* and *#SoSad* as negative.

These two tasks have been successfully tackled using supervised machine learning algorithms by representing the target tweets or words as vectors of features and using hand-crafted sentiment labels for training. A major limitation of supervised approaches is

[*] Corresponding author. E-mail: fbravoma@waikato.ac.nz.
[1] http://www.twitter.com
[2] The length limit of tweets was expanded to 280 characters in November of 2017.

2405-6456/0-1900/$35.00 © 0 – IOS Press and the authors. All rights reserved

that the annotation of words or tweets based on sentiment classes is a time-consuming and labour-intensive task.

Relying on external resources of knowledge such as labelled data from a related problem or unlabelled data has shown to be useful when labelled training data is scarce. For example, several studies have successfully used knowledge provided by polarity lexicons for the MPC task [3] or used tweets annotated by sentiment for PLI [4, 5]. Large collections of unlabelled tweets retrieved from the Twitter API[3] have also been exploited using semi-supervised [6] or representation learning [7] approaches.

Transfer learning refers to the process of improving the learning of a predictive function for a target domain $\mathcal{D}_T$ using knowledge obtained from a related source domain $\mathcal{D}_S$ [8]. This paper presents an instance aggregation framework for transferring sentiment knowledge from the word domain $\mathcal{D}_W$ to the message domain $\mathcal{D}_M$ and vice versa[4]. This transfer learning approach is useful in scenarios where either MPC or PLI needs to be solved but it is easier to obtain annotated data from the other domain. This paper extends a previous conference paper [9] and provides a more thorough and detailed report. The previous model is generalised into a framework and we use word embeddings [10] as an alternative instantiation of this framework.

Transfer learning requires both the source and target domain to be related. We motivate the relatedness between words and tweets based on three perspectives: 1) the data modelling perspective, 2) the semantic perspective, 3) and the sentiment perspective.

The data modelling perspective uses the notion of aggregation [11] for interrelating words and tweets:

- A tweet can be represented as the aggregation of the words it contains.
- A word can be represented as the aggregation of the tweets that contain it.

This perspective tells us that there is a whole-part relationship between words and tweets, in which instances of both domains can be represented as the aggregation of instances from the other.

The semantic relatedness perspective considers two linguistic theories for relating the meaning of tweets and words: 1) the principle of semantic compositionally [12], and 2) the distributional hypothesis [13].

According to the principle of semantic compositionally, the meaning of a sentence can be determined by the meaning of its lexical units (phrases and words) together with the manner in which these units are combined. This suggests that the meaning of a tweet (usually formed by a single sentence) can be inferred from the individual meanings of its words.

According to the distributional hypothesis, the meaning of a word is determined by the contexts in which it occurs. Since tweets are short messages, it is reasonable to consider a tweet as the whole context of its words. Hence, the underlying meaning of a word can be determined from the collection of tweets that contain it[5].

Tweets and words can be annotated with the same sentiment categories (e.g., positive, negative, neutral). However, the underlying cognitive task of associating instances with sentiment differs from one domain to another. While the sentiment label of a tweet corresponds to a view, attitude, or appraisal expressed by an opinion holder (which is usually the author) in the message [14], the sentiment of a word corresponds to its prior polarity (a.k.a semantic orientation or sentiment association) when the word is considered in isolation.

Sentiment can be viewed as a sub-dimension of semantics. Prior work shows that synonyms tend to have the same polarity and antonyms have the opposite one [14, 15]. Based on this, and incorporating the data modelling and semantic perspectives, we relate the sentiment of tweets and words using the following interdependence relation:

1. The polarity of a tweet is determined by the polarity of the words it contains.
2. The polarity of a word is determined by the polarity of the tweets in which it occurs.

This relation was firstly proposed in [16] in the context of larger text documents. We extend it to short informal messages.

In our proposed transfer learning framework, we use the data perspective for representing tweets and words by compatible feature vectors of the same dimensionality. The framework starts with a given vectorial representation for one domain and obtains compatible vectors for the other domain by averaging the vectors of its constituent instances. Taking the sentiment perspective into account, we expect that the aver-

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

[4] The terms “message” and “tweets” will be used interchangeably in this paper.

[5] The collection must be sufficiently large to capture the distributional properties of the word.

aged vectors will transfer sentiment information from one domain to the other.

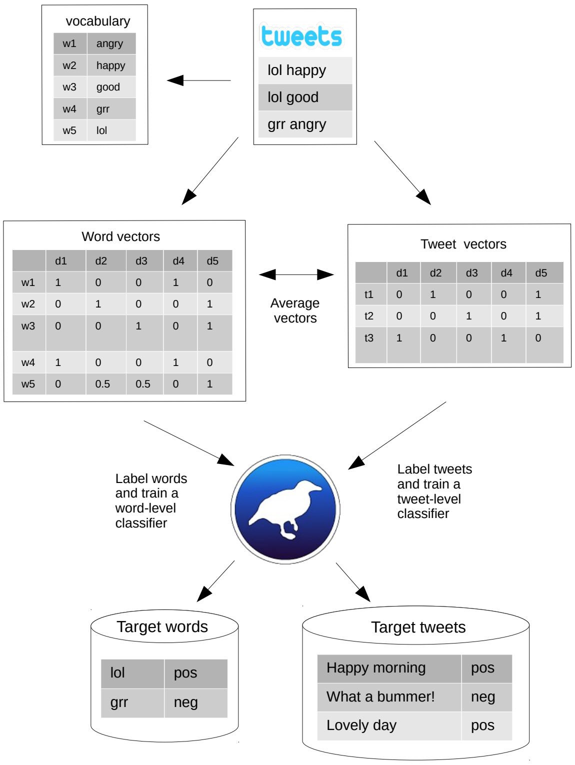
We study two models based on this framework: 1) the tweet centroid model, and 2) the word centroid model.

1. In the tweet centroid model, tweets are used as the initial representation and are represented using standard natural language processing (NLP) features such as unigrams and part-of-speech (POS) tags, and words are represented by the centroids of the tweet vectors in which they occur.
2. In the word centroid model, we first calculate word vectors from unlabelled tweets using the skip-gram embedding model [10] and represent tweets as the centroid of the word vectors.

In this way, a word-level classifier trained from a polarity lexicon can be used for classifying the sentiment of tweets (MPC). Likewise, we can train a message-level classifier from a corpus of sentiment-annotated tweets and use it for classifying words into sentiment classes (PLI). Unlabelled data also plays a crucial role for learning better word vectors as will be discussed in Section 5.

The main contribution of this paper is this new framework for transferring sentiment knowledge between words and tweets based on representing them by feature vectors of the same dimensionality. A noteworthy aspect of the framework is its simplicity; yet, despite its simplicity, it yields promising classification performance, as we show in Section 5. The framework is illustrated in Figure 1.

This article is organised as follows. Basic notations and definitions are given in Section 2. In Section 3, we provide a review of related work. The proposed transfer learning approach is described in more detail in Section 4. In Section 5, we present the experiments we conducted to evaluate the proposed approach and discuss results. The main findings and conclusions are discussed in Section 6.


vocabulary
tweets
Word vectors
Average vectors
Tweet vectors
Label words and train a word-level classifier
Label tweets and train a tweet-level classifier
Target words
Target tweets

w1	angry
w2	happy
w3	good
w4	grr
w5	lol

tweets
lol happy
lol good
grr angry

	d1	d2	d3	d4	d5
w1	1	0	0	1	0
w2	0	1	0	0	1
w3	0	0	1	0	1
w4	1	0	0	1	0
w5	0	0.5	0.5	0	1

	d1	d2	d3	d4	d5
t1	0	1	0	0	1
t2	0	0	1	0	1
t3	1	0	0	1	0

lol	pos
grr	neg

Happy morning	pos
What a bummer!	neg
Lovely day	pos

Fig. 1. Instance aggregation framework for transfer learning between words and tweets. The tweet vector dimensions correspond to the different words from the vocabulary, and the word vectors are calculated using the tweet-centroid approach, i.e., by averaging the tweet vectors of the tweets where they occur. The bird represents the Weka machine learning software, which we use in our implementation of our framework.

2. Notations and Definitions

In this section, we provide notations and definitions that will be used throughout the paper. We formalise the MPC and PLI problems, as well as the transfer learning tasks studied in this paper. Following the notation proposed in [8], a domain $\mathcal{D}$ consists of two components: a feature space $\mathcal{X}$ and a probability distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$ and each x_i is a numeric feature. Given a particular domain $\mathcal{D}$, a task $\mathcal{T}$ consists of a label space $\mathcal{Y}$ and a predictive function f that can be learned from training data consisting of pairs $\{x, y\}$ where $x \in X$ and $y \in \mathcal{Y}$. The function f can be used for predicting the corresponding label $f(x)$ of a new instance x.

In the Twitter sentiment analysis context, a word w is a unique sequence of characters taken from an alphabet Σ, and a tweet or message m is a sequence of words (limited to 140 characters) separated by special characters called delimiters (e.g., white space, punctuation symbols) that are not part of Σ.

In order to build a message domain $\mathcal{D}_M$, tweets are mapped into k-dimensional vectors $\overrightarrow{x_m}$ to form the feature space $\mathcal{X}_M$. A popular choice for building $\mathcal{X}_M$ is the vector space model [17], in which all the different words or unigrams found in the corpus are mapped into individual features. Word n-grams, which are consecutive sequences of n words, can also be used analogously. Each tweet is represented as a sparse vector whose active dimensions (dimensions that are different from zero) correspond to the words or n-grams found in the message. The values of each active dimension can be calculated using different weighting schemes, such as binary weights or frequency-based weights with different normalisation schemes.

The message-level sentiment label space $\mathcal{Y}_M$ corresponds to the different sentiment categories that can be expressed in a tweet, e.g., positive, negative, and neutral. For simplicity, we will only consider the two-class (positive and negative) case in this paper but our approach is directly applicable to multi-class problems as well. Because sentiment is a subjective judgement, the ground-truth sentiment category of a tweet must be determined by a human evaluator.

Let $\mathcal{C}$ be a corpus of tweets. We distinguish between two types of Twitter corpora: 1) unlabelled corpora $\mathcal{C}_U$, and 2) sentiment-annotated corpora $\mathcal{C}_L$.

Annotated corpora are often not available due to the high costs involved in the annotation process. Conversely, a large corpus of unlabelled public tweets $\mathcal{C}_U$ can be freely obtained from the Twitter API. Tweets restricted to a specific language, geographical region, or set of key words can also been collected for creating domain-specific collections.

Given a corpus of sentiment-annotated tweets $\mathcal{C}_L$ formed by pairs of the form $\{x, y\}$, the MPC task $\mathcal{T}_M$ consists of learning a message-level polarity classification function f_M using any supervised learning approach (e.g., SVMs, logistic regression model, Naive Bayes). The function f_M can be readily applied to any collection of unlabelled tweets.

Words can be annotated according to the same sentiment categories as messages ($\mathcal{Y}_W = \mathcal{Y}_M$) to indicate their prior sentiment. Examples of positive words are *happy* and *great*, and examples of negative ones are *sad* and *miserable*. Again, the ground-truth sentiment of a word is a subjective judgement determined by a human. We refer to a list of words annotated by sentiment as a polarity lexicon $\mathcal{L}$.

Let $\mathcal{V}$ be the vocabulary formed by the distinct words found in a corpus of tweets. A word domain $\mathcal{D}_W$ is formed by all the words in $\mathcal{V}$ represented by k-dimensional vectors $\overrightarrow{x_w}$. We consider distributional vectors as well as word embeddings in this paper. Distributional vectors [18] are used for representing lexical items such as words according to the context in which they occur in a corpus of documents or tweets. In other words, distributional models infer the meaning of a word from the distribution of the words that surround it. Word embeddings are low-dimensional continuous dense word vectors trained using neural networks [19] that have shown to perform well across several NLP tasks (e.g., named-entity recognition, chunking, parsing).

The word vectors that match a given polarity lexicon $\mathcal{L}$ can be used to form a word-sentiment training dataset. We define the PLI problem as the task $\mathcal{T}_W$ of learning a word-level classifier f_W from this dataset using supervised learning algorithms. Polarity lexicon induction is then conducted by deploying f_W on the words from the corpus that are not contained in $\mathcal{L}$.

As discussed in [8], the study of transfer learning is motivated by the idea that knowledge learned for one task can aid solving other tasks with faster or better solutions. The unified definition of transfer learning proposed in that paper is given as follows:

Definition 1 (Transfer Learning). *Given a source domain $\mathcal{D}_S$ and a learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function f_T in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.*

In the following sections, we will show how to transfer knowledge from the word domain $\mathcal{D}_W$ to solve the MPC task $\mathcal{T}_M$, and how to transfer knowledge from the message domain $\mathcal{D}_M$ to solve the PLI task $\mathcal{T}_W$. However, first, we review related work.

3. Related Work

Previous work on transfer learning for sentiment analysis focuses on adapting document-level sentiment classifiers trained on labelled reviews from a source domain, e.g., movie reviews, to a target domain where a different vocabulary is used, e.g., kitchen appliances [20].

A recursive neural tensor network for learning the sentiment of pieces of texts of different granularities, such as words, phrases, and sentences, was proposed in [21]. The network was trained on a sentiment annotated treebank[6] of parsed sentences for learning compositional vectors of words and phrases. This method is difficult to apply to Twitter data because of the lack of Twitter-specific sentiment treebanks and robust PCFG constituency parsers for Twitter [22].

[6] http://nlp.stanford.edu/sentiment/treebank.html

There is a family of models that incorporate lexical knowledge provided by opinion lexicons for training document-level sentiment classifiers. In [16], words and documents are jointly represented by a bipartite graph of labelled and unlabelled nodes. The sentiment labels of words and documents are propagated to the unlabelled nodes using regularised least squares. In [23], the term-document matrix associated with a corpus of documents is factorised into three matrices specifying cluster labels for words and documents using a constrained non-negative tri-factorisation technique. Sentiment-annotated words and documents are introduced into the model as optimisation constraints. A generative naive Bayes model based on a polarity lexicon, which is then refined using sentiment-annotated documents, is proposed in [24].

In the following subsections, we describe previous work on polarity classification of tweets and words. Afterwards, we discuss these approaches in the context of the transfer learning problems addressed in this paper.

3.1. Sentiment Classification of Tweets

Regarding the MPC task for tweets, state-of-the art solutions are based on supervised models such as logistic regression models and support vector machines trained from hand-annotated polarity corpora. Some of the features used for describing the tweets are: word *n*-grams, character *n*-grams, part-of-speech tags, word clusters trained with the Brown clustering method [25], the number of elongated words (words with one character repeated more than two times), the number of words with all characters in uppercase, presence of positive or negative emoticons, the number of individual negations, the number of contiguous sequences of dots, question marks and exclamation marks, and features derived from polarity lexicons [1, 26].

Deep learning approaches have also been adopted for Twitter sentiment analysis. A supervised learning framework that uses sentiment-specific word embeddings and hand-crafted features was developed in [27]. The word embeddings are obtained from emoticon-annotated tweets using a tailored neural network that captures the sentiment information of sentences and the syntactic contexts of words.

A convolutional neural network architecture is developed in [28]. Each tweet is represented as a matrix whose columns correspond to the words in the tweet, preserving the order in which they occur. The words are represented by dense vectors or embeddings trained from a large corpus of unlabelled tweets. The network is formed by the following layers: an input layer with the given tweet matrix, a single convolutional layer, a rectified linear activation function, a max pooling layer, and a soft-max classification layer.

The weights of the neural network are pre-trained using emoticon-annotated data, and then trained with the hand-annotated tweets. Experimental results show that the pre-training phase allows for a proper initialisation of the network's weights, and hence, has a positive impact on classification accuracy.

Recurrent neural networks with gated units such as long short-term memory networks (LSTMs)[29] and gated recurrent units (GRUs)[30] can encode sequential data of arbitrary length as fixed-size vectors while learning long temporal dependencies e.g., distant words within a sentence exhibiting a strong semantic relation. These networks have been successfully employed for sentiment classification of sentences [31], documents [32], and tweets [33]. The input passages (e.g, tweets, sentences) are initially modelled as sequences of word vectors (usually taken from pre-trained word embeddings) and fed into a recurrent neural network that maps the whole sequence into a dense vector. This vector is later fed into a fully connected multi-layer perceptron with a soft-max output layer for sentiment prediction. A hierarchical architecture is used to model entire documents in [32]. A first recurrent network is used to encode sentences using the same approach described above, and a second recurrent network is used to encode entire documents by mapping the sequence of sentence vectors obtained from the first network into a single dense vector that can be used for classification.

Distant supervision is a popular strategy for addressing the label sparsity problem of supervised models in MPC. In these methods, raw tweets gathered from the Twitter API[7] are automatically labelled into positive and negative classes using strong sentiment signals such as positive and negative emoticons, e.g., :), :([34–36], or emotional hashtags [26], e.g., #joy, #sadness. The signals are normally discarded from the content for feature extraction. However, these approaches are ill-suited to domains such as politics where emoticons or emotional hashtags are rarely used to express positive and negative opinions.

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

An emoji-based distant supervision neural network model called DeepEmoji[8] for detecting sentiment and other affective states from short social media messages was proposed in [33]. Authors collected a large corpus of 634 million tweets containing 64 different emojis. A recurrent neural network was trained to predict the emojis from the tweet's content. The network architecture is an LSTM variant formed by an embedding layer, two bidirectional LSTM layers with normal skip connections and temporal average pooling-skip connections. A transfer-learning method is proposed to fine-tune the network for any given affective detection tasks, such as the detection of emotion, sentiment, or sarcasm. The tuning process works by initially fine-tuning each layer of the network individually and then fine-tuning all the layers together. The supervision signal is taken from the target task.

Another approach for tackling MPC in Twitter is proposed in [37]. This approach is based on distant supervision and lexical prior knowledge. The authors build a graph that has users, tweets, words, hashtags, and emoticons as its nodes. A subset of these nodes is labelled by prior sentiment knowledge provided by a polarity lexicon, the known polarity of emoticons, and a message-level classifier trained with emoticons. These sentiment labels are propagated throughout the graph using random walks.

3.2. *Polarity Lexicon Induction*

A common approach for addressing the PLI task is to calculate a word-level sentiment score based on how frequently a word occurs in positive and negative messages. This measure, referred to as PMI semantic orientation, is calculated as the difference between the point-wise-mutual information (PMI) of a word occurring in positive and negative messages [38]. The message-level sentiment labels can be obtained through distant supervision [1], or using a self-training approach. In the latter case, a message-level classifier trained from a small corpus of hand-annotated tweets is used to classify a large collection of unlabelled messages from which the word-level sentiment scores are computed [4].

Another approach is to induce the lexicon by representing Twitter words from a corpus of tweets as vectors that are used together with a small group of labelled words for training a word-level polarity classifier. The resulting classifier is then deployed on the remaining unlabelled words for performing the induction. In [39], PMI-based semantic orientation was used together with other associations between words and emoticon-annotated tweets for building the classifier's feature space. In [2], state-of-the-art word embeddings such as skip-grams [10], continuous bag-of-words [10], and Glove [40] were used as features in a regression model to determine the association between Twitter words and positive sentiment. In [7], a hybrid loss function for learning sentiment-specific word embeddings was proposed. The embeddings were obtained by combining syntactic information provided by the skip-gram model [10] and sentiment information provided by emoticon-annotated tweets. In [41], words are represented by skip-gram embeddings and tweets as the sum of the word vectors appearing in them. A message-level polarity classifier is trained from emoticon-annotated tweets and deployed on the word vectors to perform lexicon induction. This approach is closely related to the word-centroid model discussed in Section 4.2 when used for solving PLI. The main difference is that we use tweets that were manually annotated by sentiment instead of relying on noisy emoticons.

3.3. *Discussion*

The results in the papers discussed above indicate that the sentiment-interdependence relation between words and messages can be helpful in the MPC and PLI tasks. Sentiment-annotated words can be used as prior knowledge for MPC, and the message-level sentiment distribution of words can be used for PLI. In this paper, we propose a unified representation that enables bidirectional transfer of sentiment classifiers between words and tweets. The main benefit of our approach is that it only requires labelled data in one of the two domains (words or messages) for transferring sentiment knowledge into the other one.

4. Transfer Learning via Instance Aggregation

Transductive transfer learning or domain adaptation [42] refers to a transfer learning approach in which the following conditions are met:

1. The source and target domains are different but related, i.e., the feature spaces of the two domains are different, or are the same but the marginal probability distributions are different.

[8] https://deepmoji.mit.edu/

2. The source and target labels are the same.
3. Labelled data is only available from the source domain.
4. Target domain unlabelled data can be exploited for training.

The instance aggregation framework we propose in this paper fits into this category.

Our framework represents instances from the word and message domains by compatible vectors. Consequently, a classifier trained on data from the source domain can be used for classifying data from the target one. The framework exploits a whole-part relationship between the two domains, i.e., instances from one domain can be modelled as the aggregation of instances from the other. This condition is satisfied for the word and tweet domains using the data modelling perspective introduced in Section 1.

One domain will act as the aggregate domain $\mathcal{D}_A$ and the other as the constituent domain $\mathcal{D}_C$, regardless of which domain is the source or the target domain. Additional inputs of the framework are: 1) the representation function $r : \mathcal{D}_C \rightarrow R^k$, that maps instances from the constituent domain into k dimensional vectors, and 2) the aggregation function $a : \{R^k_1, \ldots, R^k_n\} \rightarrow R^k$ that maps a collection of k-dimensional vectors into a single k-dimensional vector. Given these components, the transfer learning procedure can described in the following steps:

1. Represent instances from $\mathcal{D}_C$ by feature vectors of k dimensions using the representation function r.
2. Represent instances from $\mathcal{D}_A$ as collections of instances from $\mathcal{D}_C$ using the whole-part relationship between the two domains.
3. Obtain a k-dimensional vector for each instance of $\mathcal{D}_A$ by applying the aggregation function a to all its constituent instances.
4. Train a classifier f on labelled instances from the source domain.
5. Deploy f on instances from the target domain.

We provide two different implementations of this framework: 1) the tweet-centroid model (TCM) and 2) the word-centroid model (WCM).

The word and message domain can play different domain roles depending on the target task (PLI or MPC) and the transfer model (TCM or WCM). All possible combinations that may occur in practice are shown in Table 1.

The TCM and WCM models are described in the following subsections.

Transfer Task	Model	Word Domain	Message Domain
PLI	TCM	target, aggregate	source, constituent
MPC	TCM	source, aggregate	target, constituent
PLI	WCM	target, constituent	source, aggregate
MPC	WCM	source, constituent	target, aggregate

Table 1
Combinations of tasks, models and domain roles.

4.1. Tweet-Centroids for Transfer learning

The tweet centroid model is a distributional representation proposed in [43] that exploits the short nature of tweets by treating them, in their entirety, as contexts of words. This is done by representing words as the centroids of the tweets in which they occur within a corpus of tweets.

In this model, the tweet domain is used as the constituent domain ($\mathcal{D}_M = \mathcal{D}_C$) and the word domain as the aggregate one ($\mathcal{D}_W = \mathcal{D}_A$). Tweets from a given corpus $\mathcal{C}$, are mapped using the representation function r, into three types of features that have proven to be useful for sentiment analysis of tweets [1]:

1. Word unigrams (UNI): a vector space model based on unigram frequency counts.
2. Brown clusters (BWN): a vector space model based on counting the frequency of word clusters trained with the Brown clustering algorithm [25]. This algorithm produces hierarchical clusters of words by maximising the mutual information of bigrams.
3. Part-of-speech tags (POS): a vector space model based on counting the frequency of each POS tag in the message.

The words from the vocabulary $\mathcal{V}$ of the corpus are represented as the collection $\mathcal{M}(w)$ of the tweets that contain them:

$$\mathcal{M}(w) = \{m : w \in m\} \tag{1}$$

We use the average vector as the function for aggregation and obtain a tweet centroid word vector $\overrightarrow{w}$ for each word. Consequently, each word w is represented by a k-dimensional vector $\overrightarrow{w}$ in which each dimension w_j is calculated as follows:

$$w_j = \sum_{m \in \mathcal{M}(w)} \frac{x_j^{(m)}}{|\mathcal{M}(w)|} \tag{2}$$

Another interpretation of the tweet centroid model is that words are treated as the expected tweet in which they might occur.

To avoid learning spurious relationships from infrequent words, words that occur in fewer than 10 tweets are discarded ($|\mathcal{W}(w)| < 10$), and in order to reduce the dimensionality of $\mathcal{X}$, we discard vector dimensions associated with very sparse features (e.g., unigrams, Brown clusters) that are active in fewer than 10 tweets.

Transfer learning requires the source and the target tasks to be related to each other. Assuming that the sentiment perspective introudced in Section 1 is true, and the sentiment of a word is determined by the sentiment of the tweets in which it occurs, we can apply the tweet centroid model for addressing MPC and PLI by taking labels from the respective other domain.

Considering that both tweets and words reside in the same feature space, given a collection of unlabelled tweets $\mathcal{C}_U$, we can classify the sentiment of messages using a word-level classifier f_W trained with tweet centroids labelled by a polarity lexicon $\mathcal{L}$. It is important to note that the number of labelled words for training f_W is limited to the number of words from $\mathcal{L}$ occurring in $\mathcal{C}_U$. Most existing hand-annotated polarity lexicons consist of fewer than $10,000$ words [3]. This means that our method is not capable of producing training datasets larger than the size of $\mathcal{L}$, regardless of the number of unlabelled tweets that are available. Hence, we propose a modification to increase the number of labelled instances our approach produces. The modification is based on partitioning the tweet set for each word. The tweet-aggregation set $\mathcal{M}(w)$ for each word from the lexicon ($w \in \mathcal{L}$) is partitioned into smaller disjoint subsets $\mathcal{M}(w)_1, \dots \mathcal{M}(w)_z$ of a fixed size determined by a parameter p. We calculate one tweet centroid vector $\overrightarrow{w}$ for each partition labelled according to $\mathcal{L}$. As is shown in Section 5.2, this modification leads to substantial improvements when transferring sentiment knowledge from words to tweets.

The reverse transfer of sentiment knowledge is also possible. Given a message-level polarity classifier f_M trained on a corpus of tweets $\mathcal{C}_L$ annotated by sentiment, a polarity lexicon can be induced by applying f_M to the words in $\mathcal{C}_L$, simply by representing these words by the centroids of the tweets in C_L that contain them. Alternatively, considering that sentiment-annotated corpora are usually small and word-level distributional representations such as these centroids capture richer semantic information when calculated from large document corpora, it is also possible to perform the induction by applying f_M to word vectors (i.e., tweet centroids) calculated from a larger corpus of unlabelled tweets $\mathcal{C}_U$.

It is important to clarify that the message domain $\mathcal{D}_M$ and the word domain $\mathcal{D}_W$ do not have the same probability distribution and, hence, our model performs transfer learning according to the definition from [8]. The probability distribution of the tweet domain, $P(X_m)$, is formed by sparse features such as unigrams and Brown clusters, whereas the distribution of the word domain, $P(X_w)$, is formed by averaging vectors from the tweet domain, which yields dense vectors with lower variance. Moreover, the conditional distributions of the two sentiment classification tasks are not the same either. $P(Y_w|X_w)$ encodes the relation between the prior polarity of a word and its distributional representation, whereas $P(Y_m|X_m)$ represents the relation between the polarity of a message and its sparse feature vector. Hence, normally, $P(Y_w|X_w) \neq P(Y_m|X_m)$[9].

The above discussion makes it clear that the two domains are different. However, assuming that the sentiment interdependence relation between words and tweets is true, we expect them to be sufficiently associated with each other to allow the transferability of sentiment knowledge between them.

4.2. Word-Centroids for Transfer Learning

In the word-centroid model we swap the aggregate and constituent domains used in the tweet-centroid model, and use words as the constituent domain ($\mathcal{D}_W = \mathcal{D}_C$) and tweets as the aggregate domain. We use the skip-gram word embedding model [10] implemented in the *Word2vec*[10] library as the representation function r for mapping words from a corpus of tweets $\mathcal{C}$ into k-dimensional vectors.

In the *Word2vec* skip-gram method, a neural network with one hidden layer is trained for predicting the words surrounding a centre word, within a window of size s that is shifted along the input corpus. The centre and surrounding s words correspond to the input and output layers of the network, respectively, and are represented by 1-hot vectors, which are vectors of the size of the vocabulary ($|V|$), with zero values in all entries except for the corresponding word index,

[9] If we consider the partitioned version of the model, the smaller the value of the partition size p, the more similar the conditional distributions of the two domains. Indeed, if p is set to one, both distributions are the same.

[10] https://code.google.com/p/word2vec/

which receives a value of 1. Note that the output layer is formed by the concatenation of the s 1-hot vectors of the surrounding words. The hidden layer has dimensionality k, which determines the size of the embeddings (normally $k \ll |V|$). The word-embedding for each word can be obtained in two ways: 1) from the projection matrix connecting the input layer with the hidden one, or 2) from the projection matrix connecting the hidden layer with the output one. The network can be efficiently trained using two algorithms: 1) "hierarchical softmax" and 2) "negative-sampling"[11], and the rationale of the model is that words occurring in similar contexts will receive similar vectors. We use hierarchical softmax in our experiments.

In our word-centroid model, each tweet is treated as the aggregation of the words it contains and we again use the average as the aggregation function a. Hence, a tweet is represented as the centroid of the embedding vectors of the words it contains. Based on the first part of the sentiment-interdependence relation discussed in Section 1, we expect that this aggregation will transfer sentiment information from the word vectors to the tweet domain.

The word-centroid model can be used for transferring sentiment from words to tweets and vice versa analogously to the tweet-centroid model. It is noteworthy that the word and tweet vectors will have a significantly smaller dimensionality with this approach than with the tweet-centroid model. Moreover, tweet vectors will no longer be sparse but dense.

The skip-gram model uses the *min_count* parameter to avoid learning noisy patterns from infrequent words. All words whose corpus frequency is less than the value of *min_count* are discarded. We set this value to 10 in our experiments.

5. Experiments

In this section, we conduct an experimental evaluation of two proposed instantiations of our framework for transferring sentiment information. The evaluation is divided into three parts. First, we empirically study the interdependence relation between tweets and words. Second, we evaluate how to transfer sentiment labels from words to tweets. Finally, we evaluate how to induce a polarity lexicon from sentiment-annotated tweets.

5.1. The word-tweet sentiment-interdependence relation

We start by studying the sentiment-interdependence relation between tweets and words: the sentiment of tweets determines the sentiment of the words they contain, while the polarity of words determines the sentiment of the tweets that contain them.

We analyse positive and negative tweets based on the polarity of their words, and likewise, describe positive and negative words from a given polarity lexicon according to the polarity of the tweets in which they occur. We expect to observe clear differences between elements of different polarities. The annotated data we use for this is taken from the *SemEval*[12] corpus of sentiment annotated tweets and the AFINN lexicon [45] of positive and negative words.

The *SemEval* [46] corpus consists of 5232 positive tweets and 2067 negative tweets annotated by human evaluators using the crowdsourcing platform Amazon Mechanical Turk[13]. Each tweet is annotated by five Mechanical Turk workers and the final label is determined based on the majority of the labels.

The AFINN lexicon consists of 1176 positive words and 2204 negative words, annotated by Finn Årup Nielsen[14], and includes informal words commonly found in Twitter such as slang, obscene words, acronyms and Web jargon. AFINN does not include emoticons in its original version.

We describe each tweet in *SemEval* by a message-level polarity variable calculated as the difference between the number of positive and negative words from the AFINN lexicon found in the message. This variable is normalised by the total number of words in the tweet. The tweets that do not have words from the lexicon are discarded, resulting in 1638 negative and 4193 positive tweets. The median of this message-level polarity variable for negative and positive tweets is -0.04 and 0.05, respectively. The polarity of positive and negative categories is also compared using a Wilcoxon rank sum test, obtaining a p-value less than $2.2e^{-16}$. Hence, there is statistical evidence that negative tweets contain more negative words than positive ones, and likewise positive tweets are more likely to contain positive words than negative ones. These results support the first part of the tweet-word sentiment-

[11] Although negative-sampling is based on the skip-gram model, it optimises a different function [44].

[12] http://www.cs.york.ac.uk/semeval-2013/task2/

[13] http://www.mturk.com

[14] http://neuro.imm.dtu.dk/wiki/AFINN

interdependence relation: the sentiment of a tweet is associated with the cumulative polarity of its words.

We also describe each word from the AFINN lexicon by a word-level polarity variable calculated as the difference between the number of positive and negative tweets that contain it. This variable is normalised by the total number of tweets in which the word is used. To reduce the noise induced by infrequent words, we discard words occurring in fewer than three tweets, resulting in 259 positive and 250 negative words. The median of the word-level polarity for positive and negative classes is 0.76 and -0.33 respectively. We compare this variable for both sentiment classes using a Wilcoxon rank sum test and the resulting p-value is again less than $2.2e^{-16}$. Hence, there is also statistical evidence that positive and negative words occur more frequently in tweets with the same polarity than in tweets with the opposite one. These results support the second part of the tweet-word sentiment-interdependence relation: the sentiment of a word is associated with the expected sentiment of tweets in which it occurs.

The distribution of the message-level and word-level polarity variables for each corresponding sentiment category is shown in the violin plots in Figure 2.

From the plots we can observe that the interquartile range of the tweet-level polarity lies below zero for the negative class and above zero for the positive one, suggesting that tweets of different sentiment classes have different distributions when considering the sentiment of their words. Regarding the words, we can again observe that the interquartile ranges lie below and above zero for negative and positive words respectively. Note that the gap between the positive and negative interquartile range is larger than the corresponding gap in the case of tweets. We believe that this is because there is more information available for describing words according to the polarity of the tweets in which they occur than for describing tweets according to the polarity of their words. In one case, the sentiment labels of the tweets in which opinion words occur are fully given by the sentiment-annotated corpus. In the other case, we only have the polarity of the words from a tweet that match the lexicon but do not have sentiment information for the other words in the tweet.

5.2. *From opinion words to sentiment tweets*

In this subsection, we evaluate the transfer of sentiment labels from words to tweets for solving MPC. We train word-level classifiers f_W on tweet vectors calculated using the tweet-centroid model (TCM) and the word-centroid model (WCM). The vectors are trained from different collections of unlabelled tweets $\mathcal{C}_U$ and labelled according to a polarity lexicon $\mathcal{L}$.

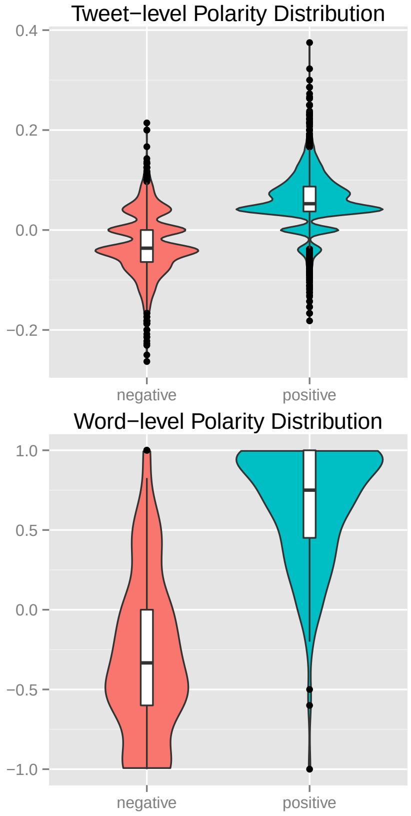


Fig. 2. Violin plots of the polarity of tweets and words.

We study the effect of partitioning the tweet-aggregation sets to increase the number of training instances obtained with the TCM model and experiment with different parameter settings of the skip-gram model in WCM.

The collections of unlabelled tweets are taken from the Edinburgh corpus [47], which is a general purpose collection of 97 million unlabelled tweets in multiple languages collected with the Twitter streaming API between November 11th 2009 and February 1st 2010. Tweets written in languages different from English are discarded, resulting in a corpus of around 50 million English tweets. We use AFINN as the polarity lexicon for the centroid labels.

In TCM, the features used for representing the tweets and the words from $\mathcal{C}_U$ are: unigrams, POS tags, and Brown clusters. In WCM, the vectors are calculated using the *Word2vec* implementation of the skip-gram model.

The tweets are lowercased, and user mentions and URLs are replaced by special tokens. The tokenisation of the tweets, the calculation of the POS tags, and the Brown clusters are taken from the **TweetNLP** library[15]. We only consider word vectors of words that are included in the AFINN lexicon.

The classification functions f_W are trained using L_2-regularised logistic regression models as implemented in LIBLINEAR[16], with the regularisation parameter C set to 1.0. We compare our models with classifiers trained using two distant supervision baselines for obtaining training instances from unlabelled corpora: the emoticon-annotation approach (EAA) and the lexicon-annotation approach (LAA).

In EAA, we use the following positive and negative emoticons for labelling tweets from the source collection: “:)”, “:D”, “=D”, “=)”, “:]”, “=]”, “:-)”, “:-D”, “:-]”, “;)”, “;D”, “;]”, “;-)”, “;-D”, and “;-]” for positive tweets and “:(”, “=(”, “;(”, “:[”, “=[”, “:-(”, “:-[”, “:’(”, “:’[”, and “D:” for negative tweets. Tweets without emoticons and tweets containing both positive and negative emoticons are discarded. The emoticons are removed from the content after labelling.

In LAA, the tweets from $\mathcal{C}_U$ are labelled using the AFINN lexicon. The tweets with at least one positive word and no negative word are labelled positive, and analogously, tweets with at least one negative word and no positive word are labelled negative. All other tweets are discarded.

It is important to recall that the training examples produced with TCM, LAA, and EAA reside in the same high-dimensional feature space. On the other hand, examples from WCM have a smaller dimensionality that is determined by the size of the skip-gram embedding.

We study several configurations of TCM. The first configuration is the basic version of TCM, in which we obtain one instance per word. The other configurations correspond to partitioned versions of TCM, in which the tweet-word sets of each word from the lexicon are partitioned into disjoints subsets of size p. The centroids are calculated from the partitions, and hence, multiple training instances are produced for words occurring in more than p tweets. The partitioning is implemented by enumerating the tweets in each word-tweet set and creating consecutive sublists of size p. The last partition of the set will be smaller than p if there is a remainder when dividing the size of the set by the value of p.

	Positive	Negative	Total
6HumanCoded	1340	949	2289
Sanders	570	654	1224
SemEval	5232	2067	7299

Table 2
Manually-annotated collections of tweets.

We also study different parameter settings of WCM. In particular, we study different values for the window size s (ranging from 1 to 15) and the number of dimensions of the embedding vector d (ranging from 50 to 800).

The evaluation of the classifiers is carried out on three manually annotated collections of tweets represented by the same features as the tweets in the training set: *SemEval*, *6HumanCoded*[17], and *Sanders*[18]. The *6HumanCoded* dataset is a collection of tweets scored according to positive and negative numeric scores by six human evaluators. The ratings are averaged and we use the difference of these scores to create polarity classes and discard messages where this difference is zero. The *Sanders* dataset is formed by tweets divided into polarity classes by a single human annotator. The number of positive and negative tweets per dataset is summarised in Table 2.

We study the average performance obtained by classifiers trained on labelled instances generated by TCM, WCM, EAA, and LAA, using ten independent subsamples of 2 million tweets from the Edinburgh corpus as the source data. The average number of positive and negative instances obtained by each model from the ten subsamples is shown in Table 3.

We can see from the table that LAA produces the largest training dataset and that the simple version of TCM and the WCM approach, which generate one instance per labelled word, produce the smallest ones. Regarding the partitioned version of TCM, we observe that the lower the value of p, the larger the number of instances produced.

By building logistic regression models from the ten training sets, we compare the average area under the ROC curve (AUC) on the three target collections of

[15] http://www.ark.cs.cmu.edu/TweetNLP/

[16] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

[17] http://sentistrength.wlv.ac.uk/documentation/6humanCodedDataSets.zip

[18] http://www.sananalytics.com/lab/twitter-sentiment/

	Avg. Positive	(%)	Avg. Negative	(%)	Avg. Total	(%)
EAA	$130,641$	(6.5%)	$21,537$	(1.1%)	$152,179$	(7.6%)
LAA	$681,531$	(34.1%)	$294,177$	(14.7%)	$975,708$	(48.8%)
TCM	1537	(0.05%)	951	(0.08%)	2488	(0.12%)
TCM (p=5)	$276,696$	(13.8%)	$149,989$	(7.5%)	$426,684$	(21.3%)
TCM (p=10)	$138,596$	(6.9%)	$75,390$	(3.8%)	$213,986$	(10.7%)
TCM (p=20)	$69,518$	(3.5%)	$38,044$	(1.9%)	$107,563$	(5.4%)
TCM (p=50)	$32,231$	(1.6%)	$17,950$	(0.9%)	$50,181$	(2.5%)
TCM (p=100)	$14,338$	(0.7%)	8357	(0.4%)	$22,695$	(1.1%)
WCM	1537	(0.7%)	955.5	(0.4%)	2492.5	(1.2%)

Table 3
Average number of positive and negative instances generated by different models from 10 collections of 2 million tweets.

tweets. We compare the results for TCM and WCM with the two baselines EAA and LAA using a paired Wilcoxon signed-rank test with a significance level of 0.05. AUC is a useful metric for comparing the performance of classifiers because it is independent of any specific value for the decision threshold. The results are given in Table 4. The outcomes of the statistical significance tests of each configuration of TCM with respect to EAA and LAA are indicated by a sequence of two symbols. Improvements are denoted by a plus (+), degradations by a minus (-), and cases where no statistically significant difference is observed by an equals (=). The baselines are also compared amongst each other.

Regarding the baselines, we observe that LAA is better than EAA on *6HumanCoded* and *SemEval* but worse on *Sanders*. The basic version of TCM is statistically significantly worse than the two baselines. We believe that this is because non-partitioned TCM generates too few training instances (Table 3). In contrast, the partitioned TCM approach achieves statistically significant improvements over the two baselines on the three datasets when p equals 10 and 20. We also observe a degradation in performance when the value of p is decreased further (p=5). This suggests a trade-off when choosing the value of p. If p is too large, TCM will generate too few training instances, and conversely, if p is too small, the instances will be calculated by averaging very few tweets, and the resulting distributional word vectors will lack contextual information.

Considering the results for WCM, we observe that the classifiers trained using this approach work better on *SemEval* than on the other two datasets. We observe a poor performance when the window size it set to 1, suggesting that embeddings trained from small context windows cannot accurately capture the sentiment of a word. We also observe poor results when setting the embedding dimension to 50. This suggests that there is a minimum number of dimensions in the semantic space required for capturing sentiment. The remaining configurations of WCM, when $s > 1$ and $d > 50$, perform very similar to each other, suggesting that there is no clear benefit from increasing the window size or the embedding dimension after a certain point. The best configurations of WCM did not beat the baselines on *6HumanCoded* and *Sanders*. However, results obtained for *SemEval*, after calibrating values of s and d, were competitive to those obtained by TCM after tuning the value of p.

We believe that the main reason that WCM is not working as well as the partitioned version of TCM for *6HumanCoded* and *Sanders* is the lack of a mechanism in WCM for increasing the number of training instances generated from a given polarity lexicon. In constrast to TCM, where word vectors are calculated by averaging tweet vectors, word vectors in WCM come from the projection matrix of a neural network. Thus, there is no clear way to partition the occurrences of a word for augmenting the number of instances with this approach.

Regarding the performance on the different datasets, we observe a systematically lower performance for *Sanders* in comparison to the other two datasets for all type of models. Considering that this is the only dataset in which labels are not obtained by averaging multiple human annotations, we believe that this dataset contains less reliable sentiment labels because it reflects the subjective judgement of a single evaluator.

The results obtained in this subsection indicate that opinion words can be successfully transferred to the message level using tweet centroids when the centroids are obtained from partitioned data. Additionally, we conclude that the partitioned tweet centroid method

	6HumanCoded		Sanders		SemEval	
EAA	0.805 ± 0.005	= -	0.800 ± 0.017	= +	0.802 ± 0.006	= -
LAA	0.809 ± 0.001	+ =	0.778 ± 0.002	- =	0.814 ± 0.000	+ =
TCM	0.776 ± 0.004	- -	0.682 ± 0.024	- -	0.779 ± 0.008	- -
TCM (*p*=5)	0.834 ± 0.002	+ +	0.807 ± 0.008	= +	0.833 ± 0.002	+ +
TCM (*p*=10)	0.845 ± 0.003	+ +	**0.817** ± 0.006	+ +	0.841 ± 0.002	+ +
TCM (*p*=20)	**0.850** ± 0.003	+ +	0.815 ± 0.011	+ +	**0.844** ± 0.003	+ +
TCM (*p*=50)	0.844 ± 0.004	+ +	0.785 ± 0.010	- +	0.836 ± 0.004	+ +
TCM (*p*=100)	0.829 ± 0.003	+ +	0.752 ± 0.019	- -	0.821 ± 0.004	+ +
WCM (*d*=50,*s*=1)	0.759 ± 0.006	- -	0.745 ± 0.007	- -	0.796 ± 0.005	- -
WCM (*d*=100,*s*=1)	0.780 ± 0.004	- -	0.752 ± 0.010	- -	0.809 ± 0.005	+ -
WCM (*d*=200,*s*=1)	0.780 ± 0.003	- -	0.751 ± 0.011	- -	0.808 ± 0.004	+ -
WCM (*d*=400,*s*=1)	0.777 ± 0.002	- -	0.750 ± 0.012	- -	0.807 ± 0.004	+ -
WCM (*d*=800,*s*=1)	0.777 ± 0.003	- -	0.751 ± 0.011	- -	0.807 ± 0.004	+ -
WCM (*d*=50,*s*=5)	0.790 ± 0.007	- -	0.760 ± 0.007	- -	0.818 ± 0.007	+ =
WCM (*d*=100,*s*=5)	0.803 ± 0.004	= -	0.770 ± 0.007	- -	0.831 ± 0.005	+ +
WCM (*d*=200,*s*=5)	0.802 ± 0.003	= -	0.771 ± 0.007	- -	0.833 ± 0.004	+ +
WCM (*d*=400,*s*=5)	0.802 ± 0.003	= -	0.771 ± 0.006	- -	0.835 ± 0.004	+ +
WCM (*d*=800,*s*=5)	0.802 ± 0.003	= -	0.770 ± 0.006	- -	0.834 ± 0.005	+ +
WCM (*d*=50,*s*=10)	0.802 ± 0.005	= -	0.763 ± 0.008	- -	0.829 ± 0.005	+ +
WCM (*d*=100,*s*=10)	0.806 ± 0.004	= -	0.774 ± 0.005	- =	0.837 ± 0.005	+ +
WCM (*d*=200,*s*=10)	0.805 ± 0.003	= -	0.776 ± 0.005	- =	0.839 ± 0.004	+ +
WCM (*d*=400,*s*=10)	0.805 ± 0.004	= -	0.777 ± 0.005	- =	0.840 ± 0.004	+ +
WCM (*d*=800,*s*=10)	0.805 ± 0.004	= -	0.778 ± 0.005	- =	0.839 ± 0.005	+ +
WCM (*d*=50,*s*=15)	0.804 ± 0.005	= -	0.764 ± 0.009	- -	0.832 ± 0.005	+ +
WCM (*d*=100,*s*=15)	0.806 ± 0.004	= =	0.775 ± 0.007	- =	0.837 ± 0.006	+ +
WCM (*d*=200,*s*=15)	0.804 ± 0.004	= -	0.776 ± 0.005	- =	0.838 ± 0.005	+ +
WCM (*d*=400,*s*=15)	0.802 ± 0.005	= -	0.776 ± 0.004	- =	0.838 ± 0.006	+ +
WCM (*d*=800,*s*=15)	0.803 ± 0.005	= -	0.776 ± 0.004	- =	0.838 ± 0.006	+ +

Table 4

Message-level Polarity Classification AUC values. Best results per column are given in bold.

is capable of extracting better information from unlabelled tweets than EAA and LAA. In relation to WCM, we observe that it is competitive to TCM for the *SemEval* dataset but not for the others.

5.3. From tweets to opinion words

We now consider whether it is possible to transfer the sentiment knowledge obtained from a sentiment-annotated corpus of tweets for creating a polarity lexicon (the PLI problem). To address this question, we train a message-level classifier f_M on a corpus of sentiment annotated tweets $\mathcal{C}_L$ and deploy it on words found in a corpus of unlabelled tweets, where the words are represented by tweet centroids and skip-gram embeddings respectively. Considering that in this task we need to have a single instance per word, we do not partition the tweet aggregation sets of the TCM model here.

Instead of calculating the word vectors using the limited amount of data available in $\mathcal{C}_L$, we calculate them from a larger corpus of unlabelled tweets $\mathcal{C}_U$ that corresponds to one of the collections of 2 million tweets used in the previous subsection. This is done for the following reasons:

1. There is empirical evidence that distributional semantic models of words tend to generalise better when calculated from large corpora [10].
2. By classifying the words from a larger corpus of unlabelled tweets we can induce the polarity of words that do not necessarily occur in the annotated corpus.

We use the three annotated collections of tweets that were previously used as testing data for training three message-level classifiers: *Sanders*, *6HumanCoded*, and *SemEval*. For TCM, we build the feature space with the same features used before: unigrams, POS tags, and Brown clusters. For WCM, we use the parameter configuration that exhibited the best performance in the previous experiment ($d = 400, s = 10$). We again use an L_2-regularised logistic regression model with the same parameters for learning the classifier. We use labelled words from the AFINN lexicon for evaluation purposes.

We compare the word-level classification AUC of a message-level classifier deployed on words represented by TCM and WCM with the AUC obtained by PMI semantic orientation (PMI-SO) [38], a popular method for inducing polarity lexicons from a corpus of polarity annotated tweets $\mathcal{C}_L$. PMI-SO corresponds to the difference between the PMI of a word with the positive class and the PMI of the same word with the negative one. Let c be a function that counts the number of times that a word w or a sentiment label y occurs in $\mathcal{C}_L$. The PMI-SO score for each word in $\mathcal{C}_L$ is calculated as follows:

$$\text{PMI-SO}(w) = log_2 \left(\frac{\text{c}(w \wedge y = pos) \times \text{c}(y = neg)}{\text{c}(y = pos) \times \text{c}(w \wedge y = neg)} \right)$$

The words classified by TCM, WCM and PMI-SO are not necessarily the same. TCM and WCM can classify all words that occur in a larger corpus of unlabelled tweets $\mathcal{C}_U$, PMI-SO can only classify the words that occur in the labelled corpus $\mathcal{C}_L$. In order to produce a fair comparison between these approaches, we compare the classification performance obtained for the intersection of the words from AFINN that are classified by the three methods. The number of positive and negative words from AFINN classified by PMI-SO for each source corpus, the number of words classified by TCM and WCM, and the number of words in the intersection, are all shown in Table 5[19].

Set of Words	Pos	Neg	Total
PMI-SO (SemEval)	522	617	1139
PMI-SO (Sanders)	196	231	427
PMI-SO (6HumanCoded)	333	352	685
TCM	961	1554	2515
WCM	966	1564	2530
PMI-SO (SemEval) ∩ TCM ∩ WCM	517	602	1119
PMI-SO (Sanders) ∩ TCM ∩ WCM	194	227	421
PMI-SO (6HumanCoded) ∩ TCM ∩ WCM	332	349	681

Table 5
Number of positive and negative words from AFINN.

The AUC scores for the intersection of words classified by PMI-SO, TCM, and WCM are displayed in Table 6. All models achieve high AUC values. This provides further evidence supporting the hypothesis that the sentiment of tweets can be transferred to the word-level. We also observe that both TCM and WCM outperform PMI-SO for solving PLI when trained on any of the three collections of sentiment annotated tweets. This is a noteworthy result, considering that PMI-SO is a widely-used approach for lexicon induction. We can also observe that classifiers trained from *6HumanCoded* and *SemEval* achieve satisfactory results on the AFINN words, and we observe a substantially lower performance for the classifier trained from *Sanders*. There is no consensus about which implementation of the instance aggregation framework works better for this task. While WCM beats TCM when trained on instances from *Sanders* and *6HumanCoded*, it exhibits a worse performance when trained on *SemEval*.

AUC			
Source Dataset	PMI-SO	TCM	WCM
Sanders	0.757	0.864	**0.892**
6HumanCoded	0.861	0.930	**0.935**
SemEval	0.858	**0.916**	0.905

Table 6
Word-level Polarity Classification Results for the AFINN lexicon. Best results per row are given in bold.

These results suggest that the performance of TCM and WCM when transferring sentiment knowledge from tweets to words can vary substantially depending on the quality of the corpus of sentiment-annotated tweets. We observe that corpora in which the labels are obtained by averaging the judgments of multiple annotators such as *6HumanCoded* and *SemEval* are preferable to corpora annotated by one single individual such as *Sanders*. The size of the corpus could also be a relevant factor, considering that *Sanders* is the smallest collection. It is worth mentioning that when an appropiate source corpus is used, the word-level performance obtained after transfer can be even better than for the reverse transfer learning task.

[19] The small variation in the number of words of TCM and WCM is caused because infrequent words are discarded differently in the two methods. While TCM considers the tweet frequency of a word, WCM considers the corpus frequency of it.

The probabilistic output of the logistic regression model applied to word vectors can be used to explore the sentiment intensities or semantic orientations of Twitter words. We calculate the log odds ratio of the positive and negative probabilities returned by the TCM logistic regression model ($\log_2(\frac{P(pos)}{P(neg)})$) for all the words found in the corpus of unlabelled tweets (here we also include words that are not part of AFINN). In this manner we obtain a sentiment score for each word. The polarity and the intensity of a word are determined by the sign and the absolute value of the score, respectively.

In Figure 3, we use word clouds to visualise the sentiment intensities of positive and negative words classified with the message-level classifier trained from the *SemEval* dataset using the TCM approach.

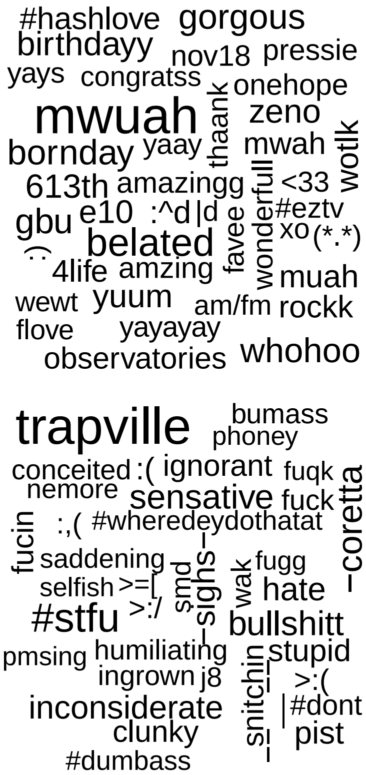


Fig. 3. Word clouds of positive and negative words obtained from a message-level classifier.

The upper word cloud corresponds to positive words for which the log odds are greater than zero ($\log_2(\frac{P(pos)}{P(neg)}) > 0$) and the size of each word is proportional to its log odds score. Analogously, in the lower word cloud, we show negative words for wich the score is less than zero and the size of the words is proportional to the score multiplied by -1. We can see that the word-level sentiment intensities transferred from message-level sentiment knowledge are plausible.

6. Conclusions

We have presented a transfer learning framework for transferring sentiment knowledge between words and tweets based on the aggregation of instances. The underlying idea is to represent both tweets and words with the same features and deploy classifiers trained from one domain on data from the other one. The source code of the model has been integrated into the *AffectiveTweets*[20] package. We studied the word-tweet sentiment interdependence relation on which the proposed framework is based, showing that the sentiment of tweets is strongly related to the sentiment of the words they contain and that the sentiment of a word is strongly related to the sentiment of the tweets in which it occurs.

We proposed two instantiations of our framework: the tweet centroid model (TCM) and the word centroid model (WCM). We observed that the partitioned version of TCM allows for accurate classification of the sentiment of tweets using a word-level classifier trained from a corpus of unlabelled tweets and a polarity lexicon of words. The partitioned TCM (with an appropriate partition size) outperformed the classification performance of the popular emoticon-based method for data labelling (EAA) and also produced better results than a classifier trained from tweets labelled using just a lexicon (LAA). Another noteworthy property of TCM is that it is, to the best of our knowledge, the first distributional model that can produce word vectors that are compatible with traditional sparse representations of tweets. Moreover, it is a flexible model: it can be used with features suitable for representing tweets. For example, paragraph vector-embeddings [48], which have shown to be powerful representations for sentences, could be trained from large corpora of unlabelled tweets and included in the message-level feature space.

The WCM approach also produced competitive results, but did not perform as well as TCM on the message-level polarity classification task. Like TCM, the WCM model is flexible in the sense that it can be

[20] AffectiveTweets is a WEKA package for analysing emotion and sentiment of tweets: https://affectivetweets.cms.waikato.ac.nz/.

used together with word-embedding models, such as continuous bag-of-words [10], Glove [40], sentiment-specific word embeddings[7], and *FastText*[21] [49].

The TCM and WCM models can be used for training message-level classifiers when no tweets annotated by sentiment are available and in domains in which emoticons are not frequently used. Considering that opinion lexicons are usually easier to obtain than corpora of sentiment-annotated tweets, transferring sentiment knowledge from the word domain using TCM or WCM can potentially significantly reduce cost when solving the message-level polarity classification problem.

Our results also show the feasibility of the reverse transfer process, where a polarity lexicon is induced by a message-level polarity classifier. We found that TCM and WCM produce more accurate lexicons than the well-known PMI-SO measure. The quality of the induced lexicon depends on the reliability of the sentiment-annotated Twitter data. An important aspect of TCM and WCM for lexicon induction is that the word vectors can be calculated from any collection of unlabelled tweets. Hence, the method can be used for creating domain-specific opinion lexicons by collecting unlabelled tweets associated with the target domain.

Our framework is also sufficiently flexible to be used with other types of sentiment-related labels for tweets or words. In future work, we will study the transferability of other sentiment-related information such as subjectivity or neutrality, numerical scores indicating sentiment strength, and multi-label emotions.

We identify two shortcoming of the aggregation functions implemented in this paper:

1. The order in which words occur in a tweet is not being considered.
2. All words and tweets respectively are considered equally important when used to model their corresponding counterpart in the whole-part relationship.

We belive that these properties impose limitations for handling complex sentiment patterns, such as negations and intensifiers.

For future work, we plan to explore other aggregation functions to handle complex sentiment compositions. In addition to the simple averaging process used in this paper, we will explore aggregation functions that weight certain tweets or words higher than others. In the latter case, we will explore information provided by the syntactic and semantic role played by a word in a sentence.

[21] https://github.com/facebookresearch/fastText

7. Acknowledgements

This work was funded by the SIF Centre for Machine Learning Applications in Education at the University of Waikato. The first author was funded by the University of Waikato Doctoral Publications Scholarship.

References

[1] S. Kiritchenko, X. Zhu and S.M. Mohammad, Sentiment analysis of short informal texts, *Journal of Artificial Intelligence Research* (2014), 723–762.

[2] S. Amir, W. Ling, R. Astudillo, B. Martins, M.J. Silva and I. Trancoso, INESC-ID: A Regression Model for Large Scale Twitter Sentiment Lexicon Induction, in: *Proceedings of the 9th International Workshop on Semantic Evaluation*, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 613–618.

[3] F. Bravo-Marquez, M. Mendoza and B. Poblete, Meta-level sentiment models for big social data analysis, *Knowledge-Based Systems* **69**(0) (2014), 86–99.

[4] L. Becker, G. Erhart, D. Skiba and V. Matula, AVAYA: Sentiment Analysis on Twitter with Self-Training and Polarity Lexicon Expansion, in: *Proceedings of the 7th International Workshop on Semantic Evaluation Exercises*, 2013, pp. 333–340.

[5] Z. Zhou, X. Zhang and M. Sanderson, Sentiment Analysis on Twitter through Topic-Based Lexicon Expansion, in: *Databases Theory and Applications*, H. Wang and M. Sharaf, eds, Lecture Notes in Computer Science, Vol. 8506, Springer International Publishing, 2014, pp. 98–109.

[6] N.F.F.D. Silva, L.F.S. Coletta and E.R. Hruschka, A Survey and Comparative Study of Tweet Sentiment Analysis via Semi-Supervised Learning, *ACM Computing Surveys* **49**(1) (2016), 15–11526.

[7] D. Tang, F. Wei, B. Qin, M. Zhou and T. Liu, Building Large-Scale Twitter-Specific Sentiment Lexicon : A Representation Learning Approach, in: *Proceedings 25th International Conference on Computational Linguistics*, 2014, pp. 172–182.

[8] S.J. Pan and Q. Yang, A Survey on Transfer Learning, *IEEE Transactions on knowledge and data engineering* **22**(10) (2010), 1345–1359.

[9] F. Bravo-Marquez, E. Frank and B. Pfahringer, From opinion lexicons to sentiment classification of tweets and vice versa: a transfer learning approach, in: *Proceedings of the 2016 IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE Computer Society, 2016, pp. 145–152.

[10] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean, Distributed Representations of Words and Phrases and their Compositionality, in: *Advances in Neural Information Processing Systems 26*, C.J.C. Burges, L. Bottou, M. Welling,

Z. Ghahramani and K.Q. Weinberger, eds, Curran Associates, Inc., 2013, pp. 3111–3119.

[11] M. Grabisch, J.-L. Marichal, R. Mesiar and E. Pap, *Aggregation Functions (Encyclopedia of Mathematics and Its Applications)*, 1st edn, Cambridge University Press, New York, NY, USA, 2009.

[12] F.J. Pelletier, The principle of semantic compositionality, *Topoi* **13**(1) (1994), 11–24.

[13] Z. Harris, Distributional structure, *Word* **10**(23) (1954), 146–162.

[14] M. Hu and B. Liu, Mining and summarizing customer reviews, in: *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, USA, 2004, pp. 168–177.

[15] S.-M. Kim and E. Hovy, Determining the sentiment of opinions, in: *Proceedings of the 20th International Conference on Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2004, pp. 1367–1373.

[16] V. Sindhwani and P. Melville, Document-Word Co-regularization for Semi-supervised Sentiment Analysis, in: *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, IEEE Computer Society, Washington, DC, USA, 2008, pp. 1025–1030.

[17] G. Salton, A. Wong and C.S. Yang, A Vector Space Model for Automatic Indexing, *Communications of the ACM* **18**(11) (1975), 613–620.

[18] P.D. Turney and P. Pantel, From Frequency to Meaning: Vector Space Models of Semantics, *Journal of Artificial Intelligence Research* **37**(1) (2010), 141–188.

[19] J. Turian, L. Ratinov and Y. Bengio, Word representations: a simple and general method for semi-supervised learning, in: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2010, pp. 384–394.

[20] X. Glorot, A. Bordes and Y. Bengio, Domain adaptation for large-scale sentiment classification: A deep learning approach, in: *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 513–520.

[21] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng and C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Vol. 1631, 2013, p. 1642.

[22] J. Foster, O. Cetinoglu, J. Wagner, J. Le Roux, J. Nivre, D. Hogan and J. van Genabith, From News to Comment: Resources and Benchmarks for Parsing the Language of Web 2.0, in: *Proceedings of 5th International Joint Conference on Natural Language Processing*, Asian Federation of Natural Language Processing, Chiang Mai, Thailand, 2011, pp. 893–901.

[23] T. Li, Y. Zhang and V. Sindhwani, A Non-negative Matrix Tri-factorization Approach to Sentiment Classification with Lexical Prior Knowledge, in: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 244–252.

[24] P. Melville, W. Gryc and R.D. Lawrence, Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification, in: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, USA, 2009, pp. 1275–1284. ISBN 978-1-60558-495-9.

[25] P.F. Brown, P.V. Desouza, R.L. Mercer, V.J.D. Pietra and J.C. Lai, Class-based n-gram models of natural language, *Computational linguistics* **18**(4) (1992), 467–479.

[26] S.M. Mohammad, S. Kiritchenko and X. Zhu, NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets, in: *Proceedings of the 7th International Workshop on Semantic Evaluation Exercises*, 2013, pp. 321–327.

[27] D. Tang, F. Wei, B. Qin, T. Liu and M. Zhou, Coooolll: A Deep Learning System for Twitter Sentiment Classification, in: *Proceedings of the 8th International Workshop on Semantic Evaluation*, Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 2014, pp. 208–212.

[28] A. Severyn and A. Moschitti, Twitter Sentiment Analysis with Deep Convolutional Neural Networks, in: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, NY, USA, 2015, pp. 959–962. ISBN 978-1-4503-3621-5.

[29] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, *Neural Computation* **9**(8) (1997), 1735–1780.

[30] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, Gated feedback recurrent neural networks, in: *International Conference on Machine Learning*, 2015, pp. 2067–2075.

[31] J. Li, T. Luong, D. Jurafsky and E. Hovy, When Are Tree Structures Necessary for Deep Learning of Representations?, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2015, pp. 2304–2314.

[32] D. Tang, B. Qin and T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1422–1432.

[33] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan and S. Lehmann, Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm, in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, Copenhagen, Denmark, September 9-11, 2017*, 2017, pp. 1615–1625.

[34] J. Read, Using emoticons to reduce dependency in machine learning techniques for sentiment classification, in: *Proceedings of the ACL Student Research Workshop*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2005, pp. 43–48.

[35] A. Go, R. Bhayani and L. Huang, Twitter Sentiment Classification using Distant Supervision, *CS224N Project Report, Stanford* (2009).

[36] A. Pak and P. Paroubek, Twitter as a corpus for sentiment analysis and opinion mining, *In Proceedings of the Seventh Conference on International Language Resources and Evaluation* (2010), 1320–1326.

[37] M. Speriosu, N. Sudan, S. Upadhyay and J. Baldridge, Twitter Polarity Classification with Label Propagation over Lexical Links and the Follower Graph, in: *Proceedings of the First Workshop on Unsupervised Learning in NLP*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 53–63.

[38] P.D. Turney, Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, in: *Proceedings of the 40th Annual Meeting on Association for Com-*

putational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002, pp. 417–424.

[39] F. Bravo-Marquez, E. Frank and B. Pfahringer, Positive, Negative, or Neutral: Learning an Expanded Opinion Lexicon from Emoticon-Annotated Tweets, in: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2015, pp. 1229–1235.

[40] J. Pennington, R. Socher and C.D. Manning, Glove: Global Vectors for Word Representation, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2014, pp. 1532–1543.

[41] G. Castellucci, D. Croce and R. Basili, Acquiring a large scale polarity lexicon through unsupervised distributional methods, in: *International Conference on Applications of Natural Language to Information Systems*, Springer Berlin Heidelberg, 2015, pp. 73–86.

[42] A. Arnold, R. Nallapati and W.W. Cohen, A comparative study of methods for transductive transfer learning, in: *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, IEEE, 2007, pp. 77–82.

[43] F. Bravo-Marquez, E. Frank and B. Pfahringer, From Unlabelled Tweets to Twitter-specific Opinion Words, in: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, NY, USA, 2015, pp. 743–746.

[44] Y. Goldberg and O. Levy, word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method, *arXiv preprint arXiv:1402.3722* (2014).

[45] F. Årup Nielsen, A new ANEW: Evaluation of a word list for sentiment analysis in microblogs, in: *Proceedings of the Workshop on 'Making Sense of Microposts': Big things come in small packages*, 2011, pp. 93–98.

[46] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter and T. Wilson, SemEval-2013 Task 2: Sentiment Analysis in Twitter, in: *Proceedings of the 7th International Workshop on Semantic Evaluation Exercises*, Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 312–320.

[47] S. Petrović, M. Osborne and V. Lavrenko, The Edinburgh Twitter corpus, in: *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 25–26.

[48] Q.V. Le and T. Mikolov, Distributed Representations of Sentences and Documents, in: *Proceedings of the 31th International Conference on Machine Learning, Beijing, China, 21-26 June 2014*, 2014, pp. 1188–1196.

[49] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, Enriching word vectors with subword information, *arXiv preprint arXiv:1607.04606* (2016).

[50] S. Baccianella, A. Esuli and F. Sebastiani, SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining, in: *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, Valletta, Malta, 2010, pp. 2200–2204.

[51] A. Bifet and E. Frank, Sentiment knowledge discovery in twitter streaming data, in: *Proceedings of the 13th international conference on Discovery science*, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 1–15.

[52] Y. Choi and C. Cardie, Adapting a Polarity Lexicon Using Integer Linear Programming for Domain-specific Sentiment Classification, in: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 590–598.

[53] R. Collobert and J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, pp. 160–167.

[54] A. Esuli and F. Sebastiani, SENTIWORDNET: A publicly available lexical resource for opinion mining, in: *In Proceedings of the 5th Conference on Language Resources and Evaluation*, 2006, pp. 417–422.

[55] L. Gatti, M. Guerini and M. Turchi, SentiWords: Deriving a High Precision and High Coverage Lexicon for Sentiment Analysis, *IEEE Transactions on Affective Computing* **7**(4) (2016), 409–421.

[56] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan and N.A. Smith, Part-of-speech tagging for twitter: Annotation, features, and experiments, in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2011, pp. 42–47.

[57] W. Guo, H. Li, H. Ji and M.T. Diab, Linking Tweets to News: A Framework to Enrich Short Text Data in Social Media, in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2013, pp. 239–249.

[58] V. Hatzivassiloglou and K.R. McKeown, Predicting the semantic orientation of adjectives, in: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, USA, 1997, pp. 174–181.

[59] L. Jiang, M. Yu, M. Zhou, X. Liu and T. Zhao, Target-dependent twitter sentiment classification, in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Association for Computational Linguistics, 2011, pp. 151–160.

[60] J. Kamps, M. Marx, R.J. Mokken and M. De Rijke, Using WordNet to Measure Semantic Orientation of Adjectives, in: *Proceedings of the International Conference on Language Resources and Evaluation*, Vol. 4, European Language Resources Association, 2004, pp. 1115–1118.

[61] E. Kouloumpis, T. Wilson and J. Moore, Twitter sentiment analysis: The good the bad and the omg!, in: *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011, pp. 538–541.

[62] B. Liu, *Sentiment Analysis and Opinion Mining*, Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers, 2012.

[63] K.-L. Liu, W.-J. Li and M. Guo, Emoticon Smoothed Language Models for Twitter Sentiment Analysis, in: *Proceedings of the National Conference on Artificial Intelligence*, Vol. 2, 2012, pp. 1678–1684.

[64] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, in: *Proceedings of Workshop at International Conference on Learning Representations*, 2013.

[65] S. Mohammad and P.D. Turney, Crowdsourcing a Word-Emotion Association Lexicon., *Computational Intelligence* **29**(3) (2013), 436–465.

[66] C. Nadeau and Y. Bengio, Inference for the generalization error, *Machine Learning* **52**(3) (2003), 239–281.

[67] K. Nigam, A. McCallum and T. Mitchell, Semi-supervised text classification using EM, *Semi-Supervised Learning* (2006), 33–56.

[68] P.J. Stone, D.C. Dunphy, M.S. Smith and D.M. Ogilvie, *The General Inquirer: A Computer Approach to Content Analysis*, MIT Press, Cambridge, MA, 1966.

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

[70] P.D. Turney and M.L. Littman, Measuring praise and criticism: Inference of semantic orientation from association, *ACM Transactions on Information Systems* **21**(4) (2003), 315–346.

[71] Y. Wilks and M. Stevenson, The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation, *Natural Language Engineering* **4**(02) (1998), 135–143.

[72] T. Wilson, J. Wiebe and P. Hoffmann, Recognizing Contextual Polarity in Phrase-level Sentiment Analysis, in: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2005, pp. 347–354.

[73] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu and B. Liu, Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis, Technical Report, Hewlett-Packard Development Company, L.P., 2011.

[74] T. Zhang, Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms, in: *Proceedings of the Twenty-first International Conference on Machine Learning*, ACM, New York, NY, USA, 2004, pp. 919–926.

[75] C. Zirn, M. Niepert, H. Stuckenschmidt and M. Strube, Fine-Grained Sentiment Analysis with Structural Features., in: *Proceedings of the 5th International Joint Conference on Natural Language Processing*, 2011, pp. 336–344.