ARTICLE IN PRESS

Knowledge-Based Systems xxx (xxxx) xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A context-aware embeddings supported method to extract a fuzzy sentiment polarity dictionary[☆]

J. Bernabé-Moreno[a], A. Tejeda-Lorente[a], J. Herce-Zelaya[a], C. Porcel[b,*], E. Herrera-Viedma[a,*]

[a] Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, Granada 18071, Spain
[b] Department of Computer Science, University of Jaén, Jaén, Spain

ARTICLE INFO

Article history:
Received 15 April 2019
Received in revised form 11 November 2019
Accepted 12 November 2019
Available online xxxx

Keywords:
Sentiment analysis
Polarity extraction
Word embeddings
Information retrieval
Contextual bias
Fuzzy polarity

ABSTRACT

The latest development in cognitive technologies are helping us understand emotions and sentiments with unprecedented precision. Polarity detection is the key enabler to sentiment analysis and typically relies on experimental dictionaries, where terms are assigned polarity scores, yet lacking contextual information and based on human inputs and conventions. In this article, we present a novel approach to automatically extract a polarity dictionary from a particular domain, the stock market, without human intervention and addressing the scaling and thresholding problem. Our approach tracks the price changes of particular stocks over time, using it as a guiding polarity value. The magnitude of the price variation for a particular stock is then attributed to the financial news about this stock in corresponding period of time and that is what we use as our working corpus. On top of that, we derive the so-called binned corpus and apply the well-known TF–IDF information retrieval techniques to compute the TF–IDF value for each term. These values are then disseminated within the neighbourhood of each term based on the embeddings-enabled cosine distance. After introducing the problem and providing the background information, we thoroughly describe our method and all the components required to implement the system. Last but not least, we assign the terms to fuzzy linguistic labels and provide a volatility metric indicating how reliable our scores are depending on their distribution of occurrences in the corpus. To show how our approach works, we implement it for the Euro Stoxx 50 from January 2018 to March 2019 and discuss the results compared with typical approaches, pointing out potential improvements for further research work.

1. Introduction

In the recent years, cognitive computing—defined as the set of software and hardware techniques mimicking the functioning of the human brain-, has experienced a substantial development [1,2]. The major cloud providers, such as Google, AWS, Microsoft and IBM, offer ready-to-use APIs for the developers' community to run these services on own data [3] and create a wider range of applications. One of the areas covered by these services is sentiment analysis, which encompasses a combination of natural language processing, text analysis, computational linguistics, and biometrics to identify, extract, and quantify in a systematic way affective states and subjective information inherent in the human communication [4–7].

Sentiment analysis has undergone a remarkable development in the last years too, becoming one of the most prolific research areas in the Natural Language Processing field [6,8]. The computation of sentiments relies heavily on the existence of polarity dictionaries, where lemmas are given a score (usually between −1 and 1) representing the contribution of words containing this lemma to the overall sentiment of the particular sentence (for example, the polarity for the word "death" according with the popular *Syuzhet* dictionary [9] is −0,75). This simplistic conception of polarity does not account for the context of the term. Continuing with the example, the word "death" in a historical context (e.g. to count the fatalities of a battle) is certainly less *negatively-loaded* than "death" in the context of journalism, when press reports breaking news about a terrorist attack in a emotional heart-breaking context. In addition, we face the so called thresholding and scaling problem. In [10] the authors show the difficulties comparing polarities given as crisp values (e.g. using

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.105236.

* Corresponding authors.
E-mail addresses: jbernabemoreno@gmail.com (J. Bernabé-Moreno), atejeda@decsai.ugr.es (A. Tejeda-Lorente), julioherce@gmail.com (J. Herce-Zelaya), cporcel@ujaen.es (C. Porcel), viedma@decsai.ugr.es (E. Herrera-Viedma).

https://doi.org/10.1016/j.knosys.2019.105236
0950-7051/

Please cite this article as: J. Bernabé-Moreno, A. Tejeda-Lorente, J. Herce-Zelaya et al., A context-aware embeddings supported method to extract a fuzzy sentiment polarity dictionary, Knowledge-Based Systems (2019) 105236, https://doi.org/10.1016/j.knosys.2019.105236.

the *Syuzhet* dictionary, we obtain −1 for "addict" and "abuse", but also for "unfit" and "sleepless"... so we have lost the possibility of comparing the terms... is "sleepless" better or worse than "addict" or "abuse") If we ask a human, probably she or he would consider "sleepless" to be "less worse" than "abuse"... but where is the threshold?

The over-reliance on these polarity scores certainly present therefore some challenges. Certainly one of the most critical ones is the fact that contextual information (leading to a contextual bias) is not captured in the polarity dictionaries (a polarity score for a lemma is immutable and not modifiable by the context). Yet, defining context-aware polarity scores for lemmas is challenging, as there is no guiding principle or systematic way of obtaining scores. In previous work [11,12], we defined methods for polarity bias modelling within a particular context, providing also a volatility score to assess how reliable our bias modelling is.

In this article, we want to go beyond polarity bias quantification and explore automatic ways of inferring polarity scores for a particular context: *finance markets*. Sentiment analysis has been extensively used to predict movements in the stock markets, to find change points, to assess the market appetite to buy or to sell and to quantify the duration of a bearish or bullish phase. The Finance markets domain, is quite appropriate to study sentiments and emotions. On one hand, a massive amount of finance related news are written everyday. News tickers provide near real time information about companies' financial health, potential events that might affect the stock course, press releases, analysts reports, product launches, etc. Specialized investment portals usually provide a news feed aggregating and tagging (e.g.: by stock symbol or company name, by index, by commodity, etc.) all potential finance news. On the other hand, we have almost real time pricing and traded volume information available in all sorts of granularities. If we assume that the choice of words, tonality, emotional load and ultimately, the sentiment is correlated to (quite important) course changes, we can also use the historical price development and the historical collection of news about a particular market entity (a symbol, a fond, etc.) to correct existing polarity scores or simply to learn new ones for the words present in the news.

The main contribution of this paper is *our approach to automatically extract a polarity dictionary from a particular domain, the stock market, without human intervention and addressing the scaling and thresholding problem*. Concretely, our contribution can be broken down into following items:

- We have created a new technique to extract news and label them with a price change magnitude, creating a weighted news collection for further processing.
- We have introduced the concept of *binned corpus*, as we are going to explain in Section 3.2
- We have re-purposed standard information retrieval techniques, such as *term frequency–inverse document frequency* to extract the guiding polarity value for each term from the binned corpus.
- We have leveraged an embeddings-based approach to compute the neighbourhood of a term and as a mechanism to disseminate guiding polarity values to the rest of the terms.
- We have transformed crisp polarity scores into fuzzy linguistic sets to make the result more generalizable and less subject to imposed thresholds and also computed the volatility related to the polarity score given the support from the domain content.

But first and foremost, with our approach we solve the three traditional issues inherent to classic polarity dictionaries based approaches:

- The scale and thresholding problem, as we provide fuzzy linguistic sets instead of crisp polarity values
- The human bias problem, as the polarity values are fully inferred without human intervention, providing on top an indicator on how reliable each particular polarity value is.
- The contextualization problem, as the polarity values are specific to our domain.

Our work is structured as follows: after presenting the rational of our attempt in the introduction, we provide the background information supporting our research. Then, we introduce important definitions we are going to use in our approach and explain thoroughly how new/corrected polarity scores are computed and mapped to linguistic fuzzy sets. Subsequently, we discuss the results obtained after applying our method in a practical case with the 50 Euro Stoxx stocks. To finish the paper, we provide the main concluding remarks and point to further research lines.

2. Background

In this section we provide the background information required to sustain our work. First we introduce the topic of polarity detection and revise the approaches to automatic polarity extraction. Then we go through the related work exploring the connection between stock prices and financial news, social media, etc. (which is one of the key assumptions for our approach to work). Then we introduce the fuzzy linguistic modelling we use to extract the fuzzy version from our intermediate crisp polarity dictionary. Finally, we review the fundamentals of word embeddings and their usage to establish relationships between terms within a corpus, which we also exploit in our method.

2.1. On polarity detection

The standard approach to *polarity detection* or *semantic orientation* uses either a pre-trained or a manually labelled polarity lexicon or dictionary. Thus, these dictionaries are at the core of any sentiment analysis related activity. One of the first examples is the dictionary created by Hu et al. [13], consisting of 6779 terms (4776 assigned to −1 and 2003 to +1), extensively used on customer reviews for opinion mining-. A commercial version issued by Daku and his co-authors, the Lexicoder [14] had a similar aim. The more recent *Syuzhet* dictionary [9] provides over 10 K entries, with scoring ranging from −1 to 1. The Positive Affect Negative Affect Scale technique (PANAS) [15] expands into the psychology domain offering a psychometric scale for detecting mood fluctuations.

The well-known SentiWordNet [16] provides a dictionary-based approach to extract sentiments. This dictionary relies on Part of Speech tagging to apply a lexical dictionary to *synsets* or synonym set groups (adjectives, nouns, verbs, and other grammatical classes). The polarity computation of a given text is an aggregation operation [8] across all the existing synsets, each one contributing with their own positive or negative affect score.

We find a lot of researchers focusing in addition on modelling happiness based on sentiments. In [17], Dodd proposed a dictionary based Happiness Index derived from the Affective Norms for English Words (ANEW). Araujo et al. in [18] suggested a method to map the happiness index to positive or negative polarity values. ANEW has been used for many other applications, such as extraction of emotional profiles for locations [19].

Some authors have approached the polarity problem from furhter angles. Thelwall et al. explored approaches to compute the sentiment strength. Their SentiStrength [20] relies on the existing *Linguistic Enquiry* and *Word Count* dictionary [21] to implement supervised and unsupervised classification methods and extract the strength of the sentiments, including polarity. Similarly, *SenticNet* [22] applies classification techniques to Natural Language Processing structures to infer the polarity for nearly 14 K concepts. The new version, *SenticNet 5* [23], implements further improvements based on deep learning techniques.

2.2. On automatic polarity extraction

As aforementioned, one of the weaknesses of classical sentiment analysis is the dependency on the quality of a polarity dictionary. As we have seen before, polarity dictionaries are typically biased, inaccurate and not context aware. Thus, many researcher have focused their work on improving the quality of the polarity dictionaries in three different ways: adapting them to the context of particular domains, defining correcting functions to the polarity value and implementing a controlled high-quality automatic way of creating polarity dictionaries. Back in 2006, Kanayama et al. introduced in [24] the notion of polar atoms and presented an coherency based approach (assuming that similar polarities tend to appear successively in context). The approach implements a redistribution of polarity values based on density and precision of coherency in a corpus. Agathangelou et al. in [25] proposed an approach for domain-specific dictionary building based on the software called NiosTo, which rather that infer polarity values from scratch, relies on existing dictionaries.

Peng and his co-authors presented in [26] an automatic sentiment dictionary generation method, called Constrained Symmetric Nonnegative Matrix Factorization (CSNMF) algorithm, to assign polarity scores to each word in the dictionary, and benchmarked the results with human-labelled dictionaries from AMT and the General Inquirer lexicon.

Back in 2006, Kanayama introduced the notion of polar atoms and presented an coherency based approach (assuming that similar polarities tend to appear successively in context). The approach implements a redistribution of polarity values based on density and precision of coherency in a corpus.

An interesting approach can be found in [27], where the authors proposed a method to readjust polarities based on the presence of *emoticons* on micro-blogs-. Basically, the approach uses the presence of *emoticons* to compute a polarity added value (extension) to the existing scores and later uses SVM to classify the sentiment word to build up the dictionary. In the same research line, Cambria et al. [28] created *Affective 2*, a language visualization and analysis system that allows for reasoning by analogy on natural language concepts. The proposal is then enhanced and generalized in [29]. In [30] the authors propose a richer deep learning powered approach to overcome the language specificity limitation inherent to Affective space vectors. Their approach builds upon the so called *Convolutional Fuzzy Sentiment Classifier* to predict the degree of a particular emotion in the Affective Space, performing in a 4 dimensional emotional space to speed up the classification performance. The recent advances in deep learning technologies have been extensively applied to the sentiment analysis. For example, Ma et al. in [31] obtained promising results augmenting the long short-term memory (LSTM) network with a hierarchical attention mechanism consisting of a target level attention and a sentence-level attention [4], extending the seminal work of [32]. Further deep learning methods, such as capsule networks, allowed for increased performance tackling sentiment classification problems. The capsule network is a structured model that solves many of the problems inherent to deep learning based text analytics. Capsules are locally invariant groups that learn to recognize the existence of visual entities and encode their properties into vectors. Capsule networks utilize a nonlinear function called squashing because capsules (groups of neurons) are represented as a vector. Capsules consider the spatial relationships between entities and learn these relationships via dynamic routing [33]. In [34] a capsule approach based on Recurrent Neural Network (RNN) has been proposed. For a given problem, one capsule is built for each sentiment category e.g., 'positive' and 'negative'. Each capsule has an attribute, a state and three modules: representation module, probability module, and reconstruction module. Based on capsule representation, the probability module computes the capsule's state probability.

The contextual bias problem in polarity detection has been addressed in the literature. In [11] the authors suggested a method to quantify and amend the contextual polarity bias using fuzzy linguistic modelling to define both the correction factor and the volatility of the inferred factor. The same method has been improved one year later introducing embeddings as a tool to capture situational and contextual interdependences [12].

2.3. On using stock markets and sentiments

Sentiment analysis has been extensively used in the context of stock markets. Our work relies on the correlation between polarity of the financial news and the stock price variation, which has been thoroughly explored to create stock price prediction models.

Bollen et al. [35] analysed how collective mood states derived from large-scale Twitter feeds show some degree to correlation with the value of the Dow Jones Industrial Average over time. For that, they leverage 2 mood tracking tools on daily Tweets, the Opinion Finder and Google Profile of Mood States, establishing the correlation between 6 mood states (Calm, Alert, Sure, Vital, Kind, and Happy) and potential price variations.

Nguyen et al. [36] developed a model that captures topics and sentiment from the social media feed simultaneously and proposed a new topic model adapting LDA (TSLDA). With their model, they proved that sentiment analysis of social media can help improve the stock prediction

The impact of financial news on stock prices is thoroughly studied by Li and his co-authors in [37,38]. In their work, they describe the creation of a sentiment space combining different polarity dictionaries (Loughran–McDonald, Harvard psychological dictionary) to enhanced a generic stock price prediction framework, showing a superior performance compared to the models just using bag of words.

Seng et al. [39] suggested an approach to develop a dictionary with grammar and multiword structure, based on sentiment orientation and score of data with added information, which in conjunction with sentiment analysis, allows to investigate the relationship between financial news and stock market volatility. The results prove a strong correlation.

2.4. Fuzzy linguistic modelling

The fuzzy linguistic approach is a tool based on the concept of linguistic variable proposed by Zadeh [40]. This theory has given very good results to model qualitative information and it has been proven to be useful in many problems.

2.4.1. The 2-Tuple fuzzy linguistic approach

The 2-Tuple Fuzzy Linguistic Approach [41] is a continuous model of information representation that allows reduction in the loss of information that typically arises when using other fuzzy linguistic approaches, both classical and ordinal [42]. To define it both the 2-tuple representation model and the 2-tuple computational model to represent and aggregate the linguistic information have to be established.

Let $\mathscr{S} = \{s_0, \ldots, s_g\}$ be a linguistic term set with odd cardinality. We assume that the semantics of labels is given by means of triangular membership functions and consider all terms distributed on a scale on which a total order is defined. In this fuzzy linguistic context, if a symbolic method aggregating linguistic information obtains a value $\beta \in [0, g]$, and $\beta \notin \{0, \ldots, g\}$, we can represent β as a 2-tuple (s_i, α_i), where s_i represents the linguistic label, and α_i is a numerical value expressing the value of the translation between numerical values and 2-tuple: $\Delta(\beta) = (s_i, \alpha)$ y $\Delta^{-1}(s_i, \alpha) = \beta \in [0, g]$ [41].

In order to establish the computational model negation, comparison and aggregation operators are defined. Using functions Δ and Δ^{-1}, any of the existing aggregation operators can be easily be extended for dealing with linguistic 2-tuples without loss of information [41]. All details can be found in our previous paper [11].

2.4.2. Multi-granular linguistic information approach

To accommodate the requirements of the different sentiment analysis methods, it is important to support different "granularity levels". Certain methods could for example only deal with yes/no values and direction only (e.g.: *"Negative Bias", "No Bias", "Positive Bias"*). Other methods might be able to incorporate higher granularity values in the aggregation operation for the sentiment computation (e.g.: *"Lowest", "Low", "Normal", "High", "Highest"*).

To enable the compatibility of sentiment analysis methods, we need to support the different granularities and provide tools to manage the multi-granular linguistic information. In [43] a multi-granular 2-tuple fuzzy linguistic modelling based on the concept of linguistic hierarchy is proposed.

A *Linguistic Hierarchy*, LH, is a set of levels $l(t, n(t))$, where each level t is a linguistic term set with different granularity $n(t)$. The levels are ordered according to their granularity, so that we can distinguish a level from the previous one, i.e., a level $t+1$ provides a linguistic refinement of the previous level t. We can define a level from its predecessor level as: $l(t, n(t)) \to l(t+1, 2\cdot n(t)-1)$. In [43] a family of transformation functions between labels from different levels was introduced. To establish the computational model we select a level that we use to make the information uniform and thereby we can use the defined operator in the 2-tuple model. This result guarantees that the transformations between levels of a linguistic hierarchy are carried out without loss of information.

2.5. On machine learning methods and word embeddings

Machine Learning has revolutionized the approach to Natural Language Processing tasks. Sentiment analysis has been one of the areas that has profited the most [44]. Socher et al. [45] implemented the so called RTNT model, exploiting the structure of the sentence to compose the single terms' sentiments in order to get the overall sentiment of the sentence. It represents the words by vectors and takes a class of tensor-multiplication-based mathematical functions to describe compositionality. The big advantage of this model is that it is very interpretable. We can visualize which words it detects to be positive or negative, and how it understands the compositions. However, we need to build an extremely large training set for every specific application. In [46], the authors explore for the use of deep convolution neural networks applied to short messages, in concrete, tweets, with astonishing results.

In the recent years, the usage of a deep learning technologies enable the representation of words as vectors and the emerging of the Words Embeddings Technologies.

The ground principle of Word embeddings (also known distributional vectors) is the continuous vectorial representations of words that follow the distributional hypothesis [47], according to which words with similar meanings tend to occur in similar context. Distributional vectors, as such, are designed to capture the characteristics of the neighbours of a term.

Distribution vectors enable arithmetic operations between words. For example, we can compute how similar 2 words in a corpus are, by using standard similarities functions, such as the cosine distance. Word embeddings are often used as the first data processing layer in a deep learning model. Embeddings are typically trained by optimizing an auxiliary objective in a large unlabelled corpus and can be used in various scenarios, such as predicting a term given its context, where the resulting distributional vectors can capture general syntactical and semantic information.

We can consider Mikolov et al. as the fathers of the distribution vectors. In 2016, these authors released two seminal papers, [48] and [49], presenting the well-known *word2vec* approach, which guarantees the scalability in the generation of word embeddings (some models available that have been trained with more than 100 billion words). Mikolov et al. presented the vectors algebra as a way to perform operations between words, as the vectors preserved the semantic consistency, for example, vec(King) − vec(woman) is close to vec(Queen). One of the most exploited features, which we extensively use in our proposal, is the support for measuring similarity between vectors, for example using measures such as *cosine similarity* or just the typical euclidean distance.

Mikolov revolutionized the word embedding with his two models: Continuous Bags Of Words, which computes the conditional probability of a target term given the context words surrounding it across a window of size k and skip-gram model, which works the other way around: predicting the surrounding context words given the central target word, being the context words assumed to be located symmetrically to the target words within a distance equal to the window size in both directions.

Following the success of word2vec, further ground-breaking algorithms approached the embeddings generation in slightly different ways. *FastText* (presented in [50]) for example learns vectors for the n-grams that are found within each word, as well as each complete word (the mean of the target word vector and its component n-gram vectors are used for training at each training step). The adjustment that is calculated from the error is then used uniformly to update each of the vectors that were combined to form the target, adding additional complexity but showing better performance in some scenarios.

GloVe (Global Vectors for words representation) [51] works similarly as word2vec with a caveat. Instead of predicting the context given word, GloVe learns by constructing a co-occurrence matrix (words x context) that basically counts how frequently a word appears in a context, applying different degrees of factorization to achieve a lower-dimension representation. In this work we are going to apply GloVe to create our embeddings in our 2 different corpora.

Word embeddings present some limitations, for example the inability to represent phrases, where the combination of two or more words (e.g., idioms like "smoke and mirrors" or named entities such as "Real Madrid') does not represent the combination of meanings of individual words. Some solutions have been researched to overcome this particular problem, such as identifying such phrases based on word co-occurrence and training embeddings for them separately [52], or directly learning n-gram embeddings from unlabelled data [53].

An additional limitation is inherent to the definition of the window for the surrounding words, which is problematic if used in tasks such as sentiment analysis [54]—semantic similarity with colliding polarities might be clustered together. The work performed by Teng and his co-authors [55] suggested a sentiment aware word embedding model based on supervised polarity incorporated into the loss function in the embeddings training phase.

The GloVe algorithm is implemented following these steps:

1. Word co-occurrence statistics gathering in a form of word co-occurrence matrix X, where each element X_{ij} represents how often word i appears in context of word j. Usually we scan our corpus in the following manner: for each term we look for context terms within some area defined by a $window\_size$ before the term and a $window\_size$ after the term. Also we give less weight for more distant words, usually using this formula: $decay = \frac{1}{offset}$
2. Define a set of soft constraints for each word pair: $w_i^T w_i + b_i + b_j = log(X_{ij})$ where w_i is the vector for the main word, w_j is the vector for the context word, b_i and b_j are scalar biases for the main and context words.
3. Define a cost function: $J = \sum_{i=1}^{V} \sum_{j=1} \varphi(X_{ij})(w_i^T w_i + b_i + b_j - log(X_{ij}))^2$ Where φ is a weighting function which help us to prevent learning only from extremely common word pairs: $\varphi(X_{ij}) = \begin{cases} (\frac{X_{ij}}{X_{max}})^{\alpha} & \text{if } X_{ij} \leq XMAX \\ 1 & otherwise \end{cases}$

3. Automatic sentiment polarity extraction

In this section we will describe how our system works to produce a fuzzy polarity dictionary extracted from the financial context. The Fig. 1 shows the process steps required to implement our approach. In the subsequent sections, we will introduce the necessary definitions and describe each module, from the data gathering until the final output.

3.1. Creation of weighted news collection

The purpose of this step is to create a collection of news with a weight assigned to proxy the overall polarity of any particular news entry. As we discussed in the introduction, our idea is to gather all news related to a specific stock and use the in-percentage daily price changes as weights, as we will see below.

There are plenty of exchanges with a large number of stocks. In order to obtain robust polarity values, we need to find stocks that have both substantial media presence and large trading volumes. For this purpose, we opted in this paper for the stocks from a well-known index, such as Euro Stoxx 50[1] (made up of fifty of the largest and most liquid stocks in the EURO zone).

Fig. 2 shows how the system for data gathering and preparation works: the stocks are used as an input for our 2 harvesting modules: the *News Harvester* pulls news related to each identified stock symbol from different finance portals (typically using RSS protocol). Likewise, the *Market Data Harvester* connects to specialized finance online portals (such as Yahoo Finance[2] to obtain the current and historical courses of the selected stocks).

The *Change Quantifier* tags those days with price changes over a particular threshold (where the price of a particular stock in absolute terms went over/under a given percentage within a particular time window compared to the price just before entering the time window). As we are going to use the magnitude of the positive or negative price change, we are going to work with different thresholds (according the frequency and magnitude of the changes, we suggest a range of thresholds from −10% or less to 10% or more in steps of 2%).

The *Matching Market/News* module selects the news that match the days labelled with any threshold value, discarding the other ones. The result is a collection of news per stock symbol, where each new is labelled with a threshold value and a sign. (e.g.: ADSGN corresponding to Adidas will have the news from the 12th of March 2018 labelled with a 10% or more, as we can see in Fig. 7, moving from 104 to 122 in 2 days, which is 14%).

If $N(K, T)$ is the whole collection of news gathered for all considered stocks K during a period of time T, we represent $n(k, t) \in N(K, T), k \in K, t \in T$ as a single news referred to a particular stock k in a particular time unit t (usually days)

Let $p(k, t)$ be the close price of the stock k in the time unit t. Let w be a time window of w units (e.g.: 2 days). A more specific selection of the w is thoroughly explained and formalized in [56]. For our purposes it is important to keep a w consistent within the same stock market and long enough to capture the impact of the news but short enough to discern important price movements. Different stock markets might work better with different values of w.

Let TH be a discrete evenly distributed finite vector of thresholds (e.g.: $TH = [2, 4, 6, 8, 10]$)

Definition 1. Weighted Bin for a particular news $n(k, t)$ issued in the time t about the stock k, is defined as the maximum value of the threshold th so that the price change of k during the next w time units is equal to or greater than th

$$WB(n(k, t), p(k, t), w, TH) = \underset{th \in TH}{\operatorname{argmax}} \times (|(p(k, t + w) - p(k, t))/100| \geq th)$$

Definition 2. Signed Weighted Bin is the Weighted Bin with a positive sign indicating a stock price increase or negative indicating a decrease:

$$sWB(n(k, t), p(k, t), w, TH) = \begin{cases} WB(n(k, t), p(k, t), w, TH), \\ \text{if } p(k, t + w) > p(k, t) \\ (-1) * WB(n(k, t), p(k, t), w, TH), \\ \text{otherwise} \end{cases} \quad (1)$$

Representing the example above in the newly introduced notations:

$n(ADSGN, '2018/03/12')$ with a window of $2days$ would have a positive weighted bin of 10% $WB(n(ADSGN, '2018/03/12'), p(ADSGN, '2018/03/12'), '2days', TH) = 10\%$

Thus, the weighted news collection is the set of all news referred to the selected stocks and their corresponding *Signed Weighted Bin*. As we are using the positive or negative price change as a proxy for the news polarity, we are interested in significant variations of the price. It can be controlled by the lower end of the TH vector (e.g.: defining a minimum price change of 4% instead of 2%). In [56], Merello et al. formalized the financial news impact problem in a timely dependent manner referred to the selection of w(in time units)

3.2. Term importance computation in binned corpus

Once the weighted collection is ready, we can proceed with the pipeline presented in Fig. 3. Each news text goes through a pre-processing step, where tokenization [57], removal of stop words [58], lematization and Part of Speech tagging (implemented with [59]) and selection of particular PoS tags (nouns, verbs, adjectives) and filtering by a minimum of occurrences (to avoid sparsity and noise)

The result is a normalized corpus, containing as many documents as relevant news identified in the subsection above. Taking it as an input, we create a new corpus with as many documents as

[1] See https://www.stoxx.com/index-details?symbol=SX5E.
[2] https://finance.yahoo.com/.

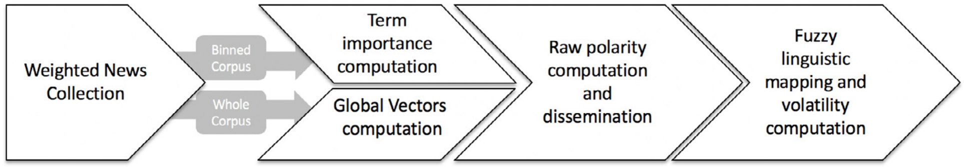

Fig. 1. Overview of the fuzzy sentiment polarity dictionary creation process.

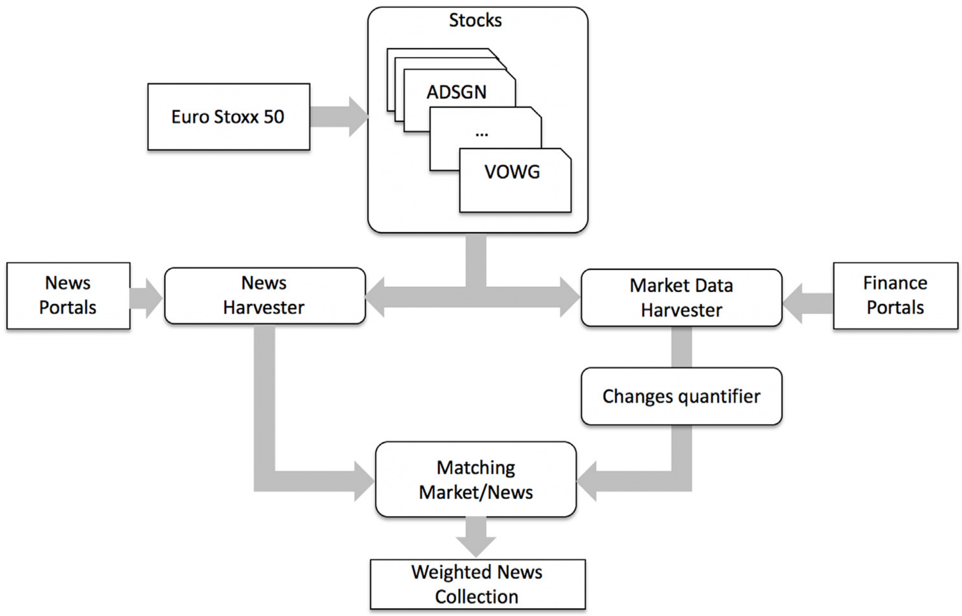

Fig. 2. Data gathering and weighted news collection creation overview.

thresholds employed in the definition of *Signed Weighted Bins*(see Definition 2). Each document is the aggregation of all the news, no matter from which stock, within the same *Signed Weighted Bin*, as explained in Fig. 4 and expressed below:

$$[n(k,t), sWB(n(k,t), p(k,t), w, TH)] \\ \rightarrow [sWB(n(k,t), p(k,t), w, TH), \\ \Xi(n(k,t), sWB(n(k,t), p(k,t), w, TH))] \quad (2)$$

where $n(k,t)$ represents a particular news about stock k in the time t, sWB has been defined in Definition 2 and $\Xi(n, th)$ represents a function that aggregates all news belonging to a particular sWB bin.

The binned corpus allows for applying the well-known algorithm TF–IDF [60], which we use to compute for each and every term, how much that term is important to that document with respect to the corpus.

Due to the nature of the stock market, smaller prices changes are more likely to happen. Thus, we can expect much higher number of news in the lower weighted bands ($\pm 2\%$, $\pm 4\%$) than in the ones reflecting higher prices changes ($\pm 8\%$, $\pm 10\%$). In order to have a proper significance when we extract the polarity, we need to establish a minimum occurrences threshold per term, which shall be proportional to the size of the weighted bin. In addition we introduce following definition to force a minimum of occurrences of a term in both corpus (binned and standard)

Definition 3 (*Polarity Computing Threshold*)**.** This is the minimum number of documents with occurrences of any term t_i in a standard corpus, so that the polarity computation makes sense. It is established for a particular Domain Corpus C and is a constant value $PCT(C) = K$.

As the binned corpus is derived from the standard corpus, the minimum occurrence condition will be only validated in the standard one.

The closer to 1 the TF–IDF value for a particular lemma in a particular signed weighted bin, the more representative is this particular lemma for this signed weighted bin. Using this relationship, we introduce the concept of guiding polarity, which combines the value signed weight bin itself and the TF–IDF of a lemma belonging to this bin:

Definition 4. Guiding Polarity $GP(t, BC)$ is the maximum absolute value obtained after multiplying the tf-idf value for a lemma l in a bin by the signed weighted value of this bin: $GP(l, BC) = |\mathrm{argmax}_{t \in TH}(tf - idf(l, t, BC)) * sWB(t)|$

Definition 5. Signed Guiding Polarity $sGP(t, BC)$ is the Guiding Polarity with the signed carried by the sWB that fulfilled the condition for Guiding Polarity

By the end of this step, we will have the positive and negative guiding polarities for the terms that are most representative, which we will use in combination with the GloVal vectors, as explained in the next section, to disseminate the polarity to other terms.

3.3. GloVal vectors computation in broader corpus

In the broader corpus, we will apply the GloVe algorithm (as explained in the Background Section 2.5) to compute the vectorial representation of our terms. As mentioned before, we stick to the pipeline of Fig. 3, applying all pre-processing steps (tokenization [57]),q40 0 removal of stop words [58], lemmatization,

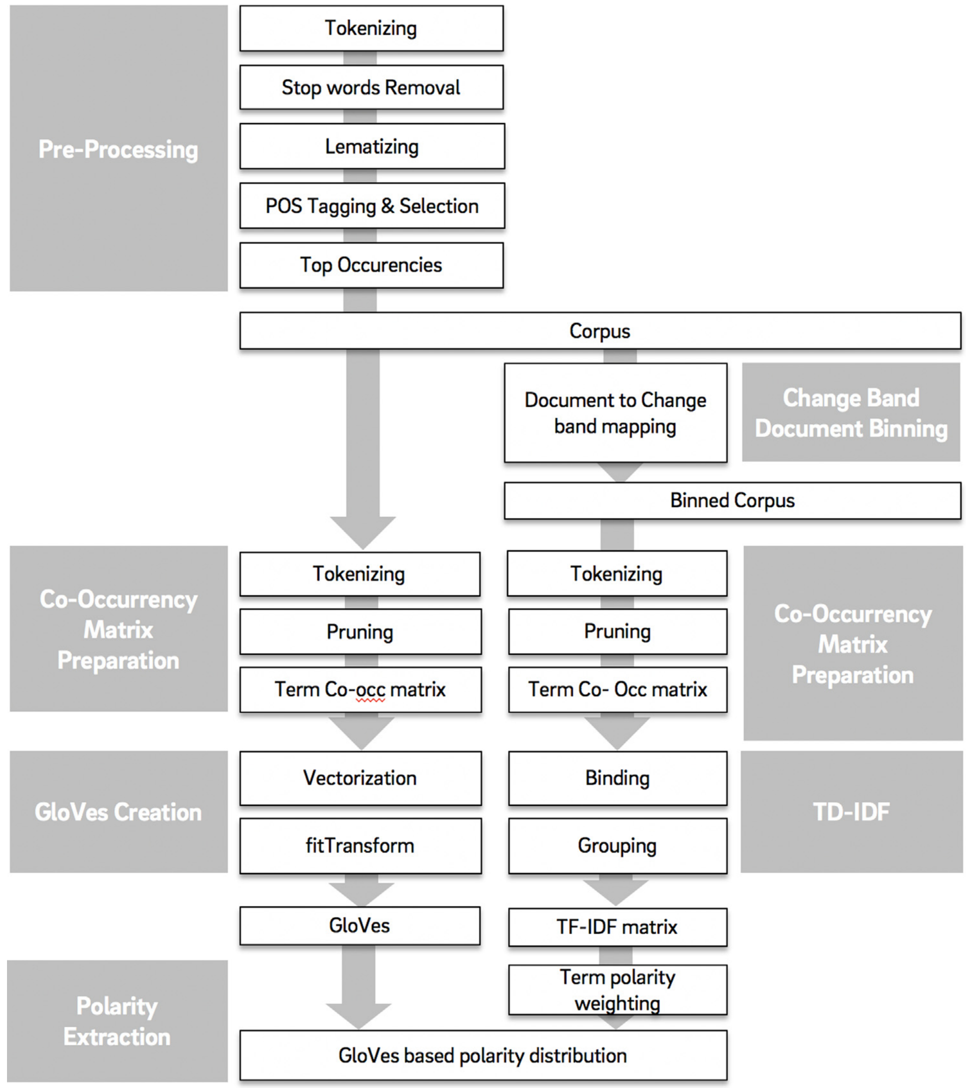

Fig. 3. Overview of the system modules to perform the context-aware polarity extraction.

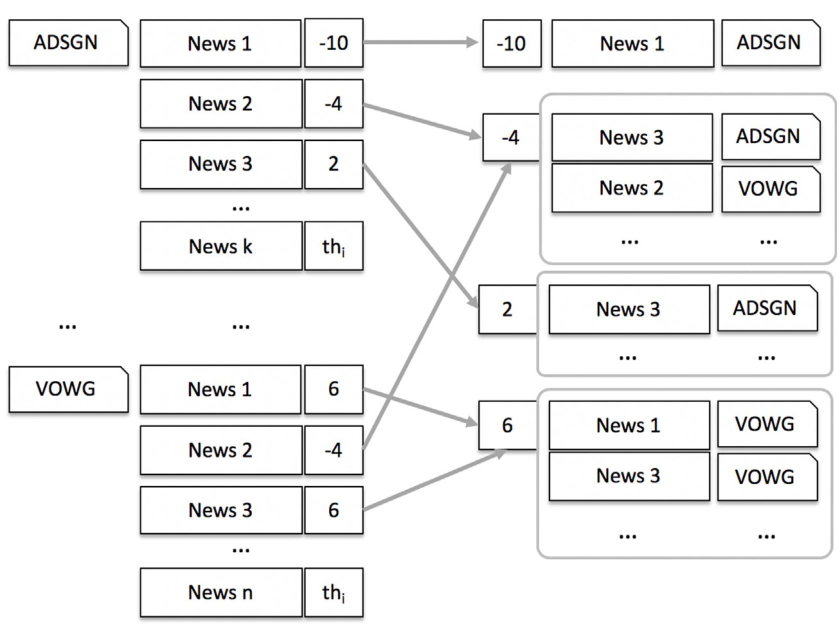

Fig. 4. Process of creation of binned corpus.

Table 1
Euro Stoxx 50 stocks used to extract our corpus and number of financial news gathered in the period of study.

	Stock	#gathered_news	Stock	#gathered_news
1	ADIDAS	210	INDITEX	20
2	AIR LIQUIDE	40	ING	210
3	AIRBUS	800	INTESA SANPAOLO	40
4	ALLIANZ	110	KERING	145
5	AMADEUS	50	KONINKLIJKE AHOLD DELHAIZE	40
6	ANHEUSER-BUSCH	300	KONINKLIJKE PHILIPS	50
7	ASML HOLDING	60	LINDE	170
8	AXA	110	LOREAL	60
9	BANCO SANTANDER	130	LOUIS VUITTON	50
10	BASF	160	MUENCHENER RUECKVERSICHERUNG	8
11	BAYER	370	NOKIA	320
12	BBVA	60	ORANGE	220
13	BMW	260	SAFRAN	70
14	BNP PARIBAS	100	SANOFI	370
15	CRH PLC	30	SAP	140
16	DAIMLER	260	SCHNEIDER ELECTRIC	50
17	DANONE	90	SIEMENS	270
18	DEUTSCHE POST	80	SOCIETE GENERALE	80
19	DEUTSCHE TELEKOM	120	TELEFONICA	200
20	ENEL	40	TOTAL	1470
21	ENGIE	300	UNIBAIL-RODAMCO-WESTFIELD	40
22	ENI	270	UNILEVER	310
23	ESSILORLUXOTTICA	5	VINCI	50
24	FRESENIUS	180	VIVENDI	180
25	IBERDROLA	50	VOLKSWAGEN	590

Part of Speech tagging, PoS tags selection (to avoid volatility we suggest to keep the most meaning carrying words, such as nouns, verbs and adjectives, but obviously our approach can be extended to any kind of Part of Speech label) and filtering by a minimum of occurrences.

Although different embeddings technologies can be applied, we have opted for Global Vectors because (a) it is very straightforward (i.e., to enforce the word vectors to capture sub-linear relationships in the vector space and therefore shows a higher performance), (b) it adds additional practical meaning into word vectors by considering the relationships between word-pair to word-pair rather than word–word and (c) it gives lower weight for highly frequent word pairs so as to prevent the dominance of meaningless stop words-like terms.

For each term, we compute the (Embeddings Neighbourhood), defined in [12] as follows:

Definition 6 (*Embedding Neighbourhood*)**.** We define the embeddings neighbourhood of a term t_i given a window length w, $EN(t_i, w)$, as the set of all terms T containing the top w terms to t_i that maximizes the cosine similarity measure, $cos(t_i, t_j) = \frac{t_i \cdot t_j}{\|t_i\| \cdot \|t_j\|}$

We will use the (Embeddings Neighbourhood) as the scope at term level to disseminate the *signed Guiding Polarity*

3.4. Guiding polarity dissemination

The step now consists of passing the guiding polarity of all terms identified in the TF–IDF procedure onto their own Embeddings Neighbourhood. Let us say k is a guiding polarity term. Let us call K the set of all terms having a signed guiding polarity. The disseminated polarity for a term l is computed as follows:

$$disPolarity(l) = \sum_k sGP(k, BC) * cossim(GloVe(l), GloVe(k)) \quad (3)$$

$l \in EN(k, w), k \in K, l \notin K$, Where k represents the set of all signed guiding polarity terms in whose neighbourhood the term l is present, $GloVe(l)$ and $GloVe(k)$ the vectorial representation of l and k respectively, and w the size of the window to define the scope of the neighbourhood (constant)

Our raw polarity dictionary is the union of signed Guiding Polarities to the disseminated Polarities.

3.5. Fuzzy linguistic mapping and volatility computation

In the previous Section 2.4, we introduced the fundamentals of fuzzy linguistic modelling and defined the 2-tupla based supporting arithmetic operations to enable the computing of sentiment analysis tasks. We now need to map the polarity values in the raw polarity dictionary obtained in the step before to linguistic labels.

In order to provide a sense of how much evidence is behind the polarity definition of a particular term, we define a measure for the stability (as opposed to volatility), based on both number of occurrences of the term in the corpus. Thus, the user of the polarity dictionary can have the choice of disregard volatile polarities.

Definition 7. Polarity Stability This is an indicator for how stable the polarity computation for a particular term is. The minimum value can be the imposed as $PCT(C)$ (as explained in Definition 7 and the maximum of $\#C$). To standardize this value, we define a normalizing function ε, defined as $\varepsilon : [PCT(C), \#C] \longrightarrow [0, 1]$, which makes the Polarity Stability value range between 0 and 1:

$$PS(t_i, C) = \varepsilon(\frac{\#M}{\#C}) \quad (4)$$

where M represents the set of documents in the standard corpus, where the term t_i is present and C the set of all documents in the Corpus.

For both cases, we are going to use different label sets (S_1, S_2) selected from a LH [43]:

- **Polarity Domain Value** of a term in a our context $PDV(t)$, which is assessed in S_1.
- **Polarity Supporting Indicator** applied to the previous indicator $PSV(t)$, which is assessed in S_2.

Although this framework guarantees the flexibility in the choice of the LH, we suggest using a 2 level LH with 3 and 5 labels each one for the *Bias Model stability indicator* and a 2 level LH with 5 and 9 labels for the *Polarity Domain Value* itself. Our suggestion is motivated by the intent of making it more tangible for the reader, but the choice of (S_1, S_2) remains generic and shall be

Please cite this article as: J. Bernabé-Moreno, A. Tejeda-Lorente, J. Herce-Zelaya et al., A context-aware embeddings supported method to extract a fuzzy sentiment polarity dictionary, Knowledge-Based Systems (2019) 105236, https://doi.org/10.1016/j.knosys.2019.105236.

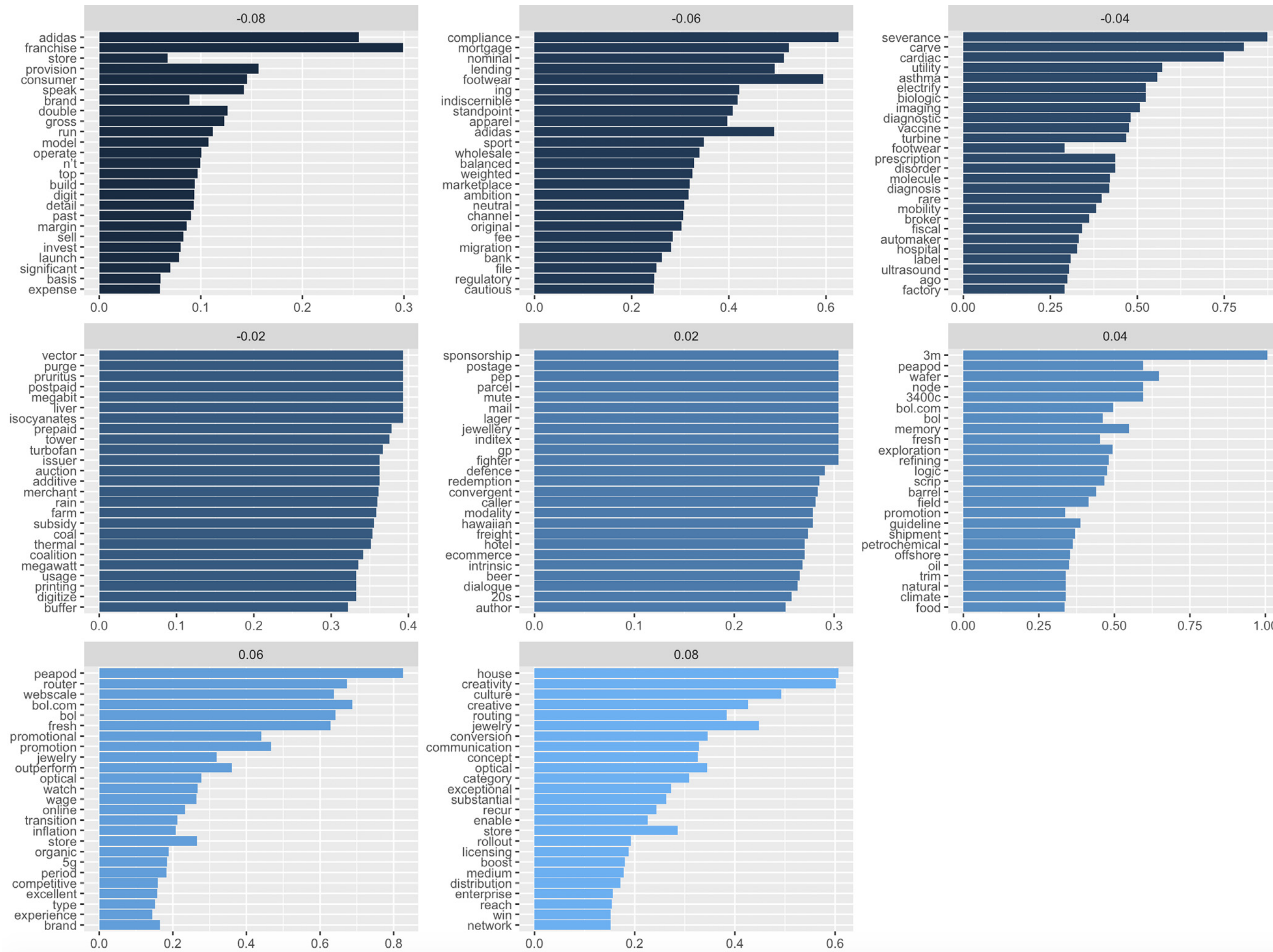

Fig. 5. Top terms in the TF–IDF step.

taken depending on the nature of the problem or convenience for further operations.

The *Polarity Domain Value* in combination with the *Polarity Supporting indicator* constitutes our context-aware fuzzy sentiment polarity dictionary D:

$D \equiv [t, PDV(t), PSI(t)]$

4. Experimentation

To implement our approach, we chose the stocks listed in the EuroStoxx 50 index[3] (composed by fifty of the largest and most liquid stocks in the EURO zone), because of the trading volumes, financial news richness and variety of industries. The Table 1 shows the concrete stocks and the number of financial news we have gathered between Jan 2018 and March 2019.

In Fig. 7 we show, taken Adidas as example, how the different price changes defining the weighted bins manifest. We have defined a time window of 2 days to register the price change, following the recommendations of [61] about news lags and delays. The period of time we have chosen presents enough price variations to support the analysis. The higher we set the threshold, the less occurrences we observe (for example, in Table 2 we just see one occurrence for a positive 10% price variation, no one for a -10%, but as we go down to $\pm 8\%$, $\pm 6\%$ up to $\pm 2\%$, we start having almost 2 K occurrences in both positive and negative bins).

Table 2
Number of news per weighted bin. E.g. positive 0,02 bin has a total of 1874 news, while the negative 0,08 bin only 7 news.

Weighted bin	+	−
0.02	1874	1893
0.04	240	324
0.06	33	43
0.08	10	7
0.1	4	0

The news assigned to the different price variations bins have undergone the pre-processing routing explained in Fig. 3 (tokenizing, stop words removal, lemmatizing and PoS Tagging). For our evaluation we just selected *nouns*, *verbs* and *adjectives*, as those are typically the highest contributors to the sentiment of a sentence. The result is a fully normalized corpus with one document per financial news gathered. Applying the formula (2), we created the binned corpus and applied TF–IDF to obtain the *signed guiding polarities*. In the Fig. 5 we can see the top 25 terms per bin visualized.

To proceed with the polarity dissemination, we applied the GloVe algorithm to create the global vectors and compute the Embeddings Neighbourhood (as explained in 3.3). In Fig. 6 we can see for example, the extended neighbourhood for the terms *retail* and *compliance*.

After aggregating both *signed guiding polarities* and *disseminated polarities*, we apply the fuzzy linguistic mapping assigning

[3] See https://www.stoxx.com/index-details?symbol=SX5E.

Please cite this article as: J. Bernabé-Moreno, A. Tejeda-Lorente, J. Herce-Zelaya et al., A context-aware embeddings supported method to extract a fuzzy sentiment polarity dictionary, Knowledge-Based Systems (2019) 105236, https://doi.org/10.1016/j.knosys.2019.105236.

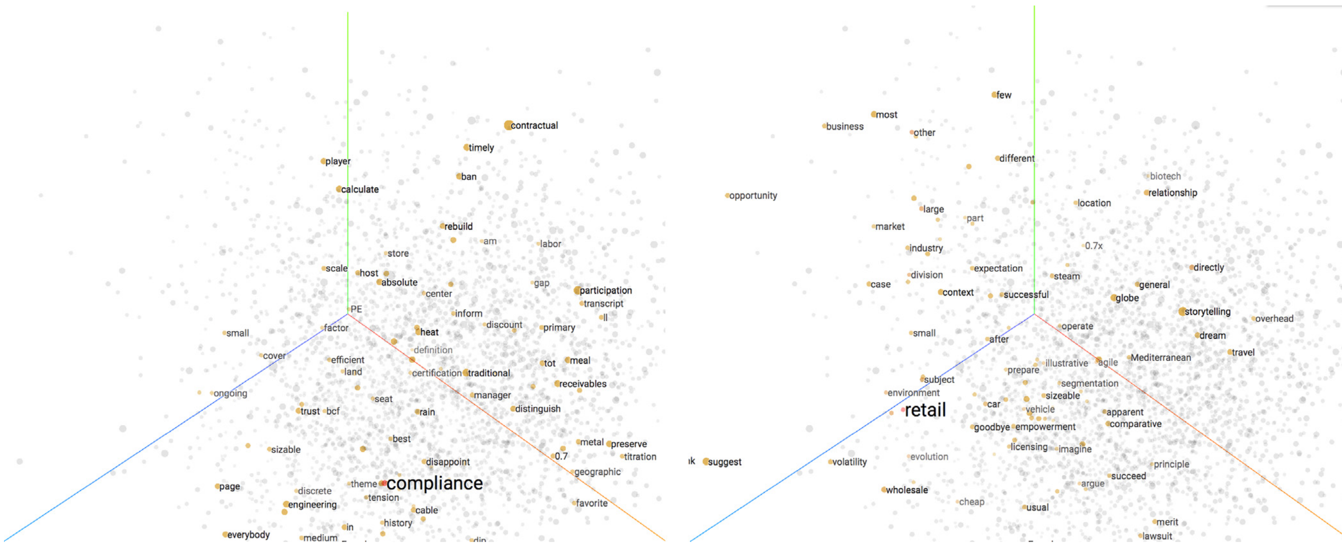

Fig. 6. Embeddings visualization for "compliance" and "retail".

Table 3
Top positive terms in the fuzzy polarity dictionary.

	Term	Maxpolarity	Maxpolarityfuzzy	Support	Supportfuzzy
1	Peapod	0.07	Very Strong positive	0.00	Very weak
2	Directly	0.07	Very Strong positive	0.00	Very weak
3	Meal	0.06	Very Strong positive	0.01	Very weak
4	Northeast	0.06	Very Strong positive	0.01	Very weak
5	bol.com	0.06	Very Strong positive	0.01	Very weak
6	Bol	0.06	Very Strong positive	0.01	Very weak
7	Nationality	0.06	Very Strong positive	0.01	Very weak
8	Fresh	0.05	Very Strong positive	0.01	Medium
9	Globe	0.05	Very Strong positive	0.02	Medium
10	Sportswear	0.05	Very Strong positive	0.00	Very weak
11	Jewellery	0.05	Very Strong positive	0.02	Medium
12	House	0.05	Very Strong positive	0.03	Strong
13	Mall	0.05	Very Strong positive	0.01	Weak
14	Creativity	0.05	Very Strong positive	0.01	Weak
15	Owned	0.04	Very Strong positive	0.01	Medium
16	Shelf	0.04	Very Strong positive	0.01	Weak
17	Relationship	0.04	Very Strong positive	0.17	Very Strong
18	Optical	0.04	Very Strong positive	0.02	Medium
19	Router	0.04	Very Strong positive	0.01	Weak
20	Compensation	0.04	Very Strong positive	0.13	Very Strong

a linguistic label to each polarity value. For our implementation, we opted for a level 5 label set, as explained in 3.5 with the labels *Almost non-existent, Slight, Medium, Strong, Very Strong* for both positive and negative polarities, to obtain the *Polarity Domain Values*

To complete our fuzzy polarity dictionary, the *Polarity Supporting Indicator* for each term is computed (as explained in Section 3.5). For this purpose, we use a different level 5 label set *Very weak, Weak, Medium, Strong, Very Strong*

In the Tables 3 and 4 we show the terms with the highest and the lowest fuzzy polarity. As we can also see, the *Polarity Supporting Indicator* helps understanding the reliability of the inferred polarities.

In Table 5 we provide the distribution of *Polarity Supporting Indicator* labels by *Polarity Domain Value* label. As we can observed, the are quite balanced.

The entire dictionary can be downloaded from https://bit.ly/2XdkyqQ

5. Concluding remarks

In this article, we introduced a novel approach to automatically extract a polarity dictionary using the stock market as the reference domain in a fully automated way (no human intervention to define polarities required).

Our system identifies price changes of particular stocks over time, using them as a guiding polarity value. The magnitude of the price variation for a particular stock is then attributed to the financial news about this stock in corresponding period of time and that is what we use as our working corpus. Using this domain corpus as reference, we build the so called *binned corpus* and apply the TF–IDF algorithm to compute the TF–IDF value for each term obtaining the signed guiding polarities. We then disseminate these values within the neighbourhood of each term based on the embeddings-enabled cosine distance. Last but not least, we map the terms to fuzzy linguistic labels and provide a supporting indicator to indicate how reliable our scores are depending on its distribution of occurrences in the corpus.

To show how our approach works, we implement it for the Euro Stoxx 50 from January 2018 to March 2019, discuss the results and made the fuzzy polarity dictionary available.

Our approach solves 3 typical issues inherent to the classic approaches to building polarity dictionaries:

- The scale and thresholding problem, as we provide fuzzy linguistic sets instead of crisp polarity values.

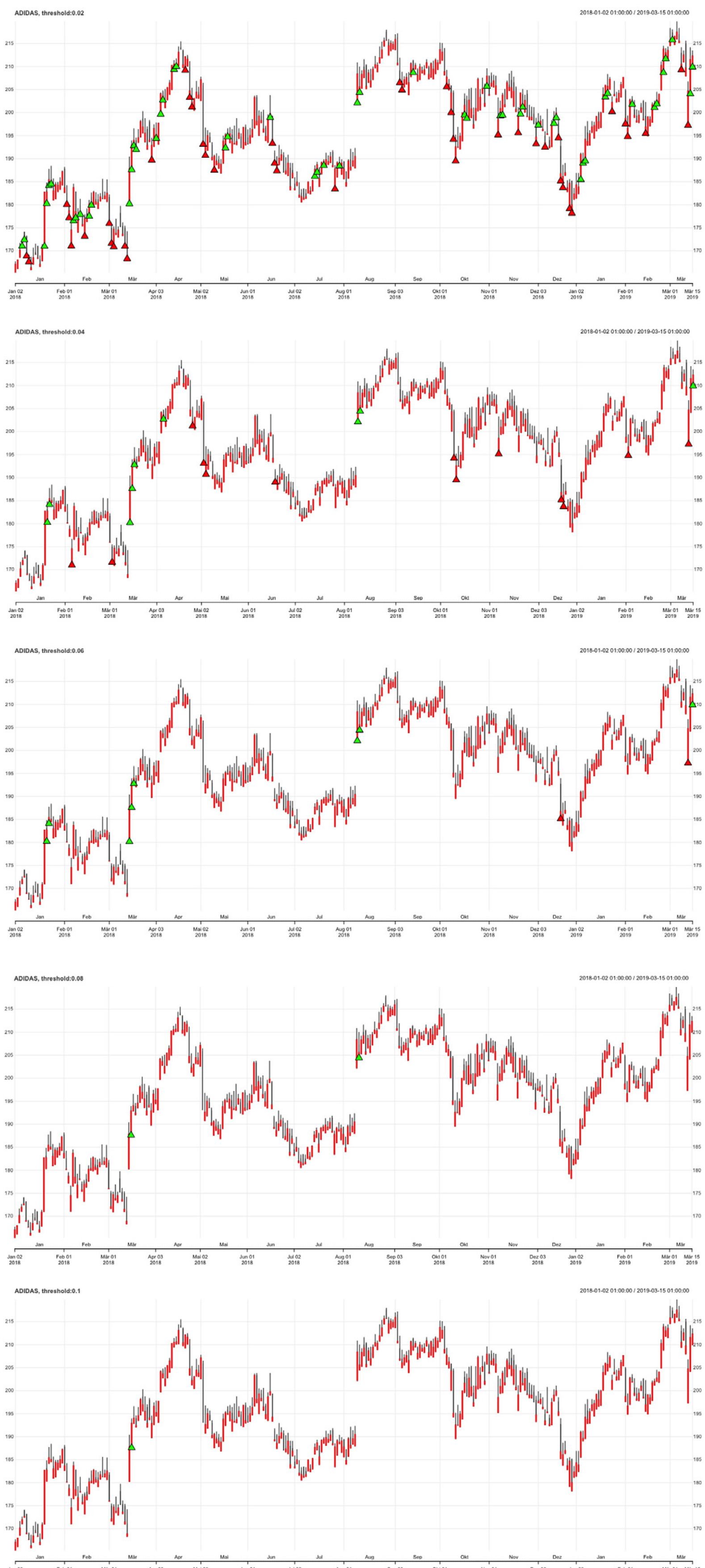

Fig. 7. Adidas course from 2018 with different change thresholds (0.02, 0.03, 0.04, 0,08, 0.1) indicating positive changes in green and negative ones in red.

Please cite this article as: J. Bernabé-Moreno, A. Tejeda-Lorente, J. Herce-Zelaya et al., A context-aware embeddings supported method to extract a fuzzy sentiment polarity dictionary, Knowledge-Based Systems (2019) 105236, https://doi.org/10.1016/j.knosys.2019.105236.

Table 4
Top negative terms in the fuzzy polarity dictionary.

	Term	Maxpolarity	Maxpolarityfuzzy	Support	Supportfuzzy
1	Jersey	−0.08	Very Strong negative	0.01	Very weak
2	School	−0.07	Very Strong negative	0.01	Very weak
3	Newness	−0.07	Very Strong negative	0.01	Very weak
4	Footwear	−0.06	Very Strong negative	0.01	Weak
5	Harm	−0.06	Very Strong negative	0.01	Weak
6	Sport	−0.06	Very Strong negative	0.02	Medium
7	Scalability	−0.06	Very Strong negative	0.01	Weak
8	Overhead	−0.06	Very Strong negative	0.02	Medium
9	Adidas	−0.05	Very Strong negative	0.02	Medium
10	Football	−0.05	Very Strong negative	0.01	Weak
11	Nominal	−0.05	Very Strong negative	0.02	Medium
12	Franchise	−0.04	Very Strong negative	0.04	Strong
13	Den	−0.04	Very Strong negative	0.01	Very weak
14	Compliance	−0.04	Very Strong negative	0.01	Medium
15	Rolling	−0.04	Very Strong negative	0.02	Medium
16	Headcount	−0.04	Very Strong negative	0.02	Medium
17	Apparel	−0.04	Very Strong negative	0.02	Medium
18	Community	−0.04	Very Strong negative	0.05	Strong
19	Replicate	−0.04	Very Strong negative	0.01	Medium
20	Mortgage	−0.04	Very Strong negative	0.01	Medium

Table 5
Distribution of supporting labels by polarity labels.

	Medium	Strong	Very Strong	Very weak	Weak
Almost non-existent negative	52	35	45	48	42
Almost non-existent positive	66	38	46	93	58
Medium negative	55	71	66	29	22
Medium positive	48	59	71	27	28
Slight negative	50	46	64	82	65
Slight positive	33	44	69	25	28
Strong negative	69	52	24	38	49
Strong positive	60	50	36	26	23
Very Strong negative	37	28	10	38	24
Very Strong positive	50	38	24	43	40

- The human bias problem, as the polarity values are fully inferred without human intervention, providing on top an indicator on how reliable each particular polarity value is.
- The contextualization problem, as the polarity values are specific to our domain.

Further research work could focus on the impact of using of n-grams instead of mono-grams as well as the extension to further Part of Speech label (adverbs, etc.). In addition, techniques to transfer the polarity dictionary to a different domain might also pave the way towards a multi-domain generic approach. Last but not least, we had like to point to all the operationalization of the polarity dictionary to compute sentiment using fuzzy linguistic arithmetic operations.

Acknowledgement

This paper has been developed with the FEDER, Spain financing under Project TIN2016-75850-R

References

[1] R. High, The Era of Cognitive Systems: An Inside Look At IBM Watson and How It Works, IBM Corporation, Redbooks, 2012.
[2] K. Hwang, M. Chen, Big-data analytics for cloud, IoT and cognitive computing, John Wiley & Sons, 2017.
[3] A. Koneru, N.B.N.S.R. Bhavani, K.P. Rao, G.S. Prakash, I.P. Kumar, V.V. Kumar, Sentiment analysis on top five cloud service providers in the market, in: 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), IEEE, 2018, pp. 293–297.
[4] O. Appel, F. Chiclana, J. Carter, H. Fujita, A hybrid approach to the sentiment analysis problem at the sentence level, Knowl.-based Syst. 108 (2016) 110–124.
[5] O. Appel, F. Chiclana, J. Carter, H. Fujita, A consensus approach to the sentiment analysis problem driven by support-based iowa majority, Int. J. Intell. Syst. 32 (9) (2017) 947–965.
[6] O. Appel, F. Chiclana, J. Carter, H. Fujita, Successes and challenges in developing a hybrid approach to sentiment analysis, Appl. Intell. 48 (5) (2018) 1176–1188.
[7] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Human Lang. Technol. 5 (1) (2012) 1–167.
[8] O. Appel, F. Chiclana, J. Carter, H. Fujita, Cross-ratio uninorms as an effective aggregation mechanism in sentiment analysis, Knowl.-Based Syst. 124 (2017) 16–22.
[9] M.L. Jockers, Revealing sentiment and plot arcs with the syuzhet package, 2015, http://www.matthewjockers.net/2015/02/02/syuzhet/.
[10] A. Devitt, K. Ahmad, Sentiment polarity identification in financial news: A cohesion-based approach, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 2007, pp. 984–991.
[11] J. Bernabé-Moreno, A. Tejeda-Lorente, C. Porcel, E. Herrera-Viedma, A fuzzy linguistics supported model to measure the contextual bias in sentiment polarity, in: Advances in Fuzzy Logic and Technology 2017, Springer, 2017, pp. 199–210.
[12] J. Bernabé-Moreno, A. Tejeda-Lorente, C. Porcel, E. Herrera-Viedma, An Embeddings Based Fuzzy Linguistics Supported Model to Measure the Contextual Bias in Sentiment Polarity, in: New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 17th International Conference SoMeT_18, Granada, Spain, 26-28 September 2018, pp. 735–748, RLhttps://doi.org/10.3233/978-1-61499-900-3-735.
[13] M. Hu, B. Liu, Mining opinion features in customer reviews, in: AAAI, Vol. 4, 2004, pp. 755–760.
[14] M. Daku, S. Soroka, L. Young, Lexicoder, version 2.0 (software), McGill University, Montreal, Canada, 2011.
[15] D. Watson, L.A. Clark, A. Tellegen, Development and validation of brief measures of positive and negative affect: the panas scales, J. Personal. Soc. Psychol. 54 (6) (1988) 1063.
[16] S. Baccianella, A. Esuli, F. Sebastiani, Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining, in: LREC, Vol. 10, 2010, pp. 2200–2204.
[17] P.S. Dodds, C.M. Danforth, Measuring the happiness of large-scale written expression: Songs, blogs, and presidents, J. Happiness Stud. 11 (4) (2010) 441–456.
[18] M. Araújo, P. Gonçalves, M. Cha, F. Benevenuto, Ifeel: a system that compares and combines sentiment analysis methods, in: Proceedings of the 23rd International Conference on World Wide Web, ACM, 2014, pp. 75–78.

[19] J. Bernabé-Moreno, A. Tejeda-Lorente, C. Porcel, H. Fujita, E. Herrera-Viedma, Emotional profiling of locations based on social media, Procedia Comput. Sci. 55 (2015) 960–969.
[20] M. Thelwall, Heart and soul: Sentiment strength detection in the social web with sentistrength, Proc. CyberEmot. (2013) 1–14.
[21] J.W. Pennebaker, M.E. Francis, R.J. Booth, Linguistic inquiry and word count: liwc 2001, Mahway: Lawrence Erlbaum Assoc. 71 (2001) (2001) 2001.
[22] E. Cambria, S. Poria, R. Bajpai, B. Schuller, Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives, in: The 26th International Conference on Computational Linguistics (COLING), Osaka, 2016.
[23] E. Cambria, S. Poria, D. Hazarika, K. Kwok, Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[24] H. Kanayama, T. Nasukawa, Fully automatic lexicon expansion for domain-oriented sentiment analysis, in: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2006, pp. 355–363.
[25] P. Agathangelou, I. Katakis, F. Kokkoras, K. Ntonas, Mining domain-specific dictionaries of opinion words, in: International Conference on Web Information Systems Engineering, Springer, 2014, pp. 47–62.
[26] W. Peng, D.H. Park, Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization, in: Fifth International AAAI Conference on Weblogs and Social Media, 2011.
[27] X. Hao, Y. Jia, Q. Gu, An automatic construction approach for sentiment dictionary based on weibo emoticons, in: 2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018), Atlantis Press, 2018.
[28] E. Cambria, J. Fu, F. Bisio, S. Poria, Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[29] E. Cambria, Affective computing and sentiment analysis, IEEE Intell. Syst. 31 (2) (2016) 102–107.
[30] I. Chaturvedi, R. Satapathy, S. Cavallari, E. Cambria, Fuzzy commonsense reasoning for multimodal sentiment analysis, Pattern Recognit. Lett. 125 (2019) 264–270.
[31] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[32] D. Cao, K. Yu, Deep attentive structured language model based on LSTM, in: International Conference on Intelligent Science and Big Data Engineering, Springer, 2017, pp. 169–180.
[33] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, in: Advances in Neural Information Processing Systems, 2017, pp. 3856–3866.
[34] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, Z. Zhao, Investigating capsule networks with dynamic routing for text classification, 2018, arXiv preprint arXiv:1804.00538.
[35] J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market, J. Comput. Sci. 2 (1) (2011) 1–8.
[36] T.H. Nguyen, K. Shirai, Topic modeling based sentiment analysis on social media for stock market prediction, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Vol. 1, 2015, pp. 1354–1364.
[37] X. Li, H. Xie, L. Chen, J. Wang, X. Deng, News impact on stock price return via sentiment analysis, Knowl.-Based Syst. 69 (2014) 14–23.
[38] X. Li, X. Huang, X. Deng, S. Zhu, Enhancing quantitative intra-day stock return prediction by integrating both market news and stock prices information, Neurocomputing 142 (2014) 228–238.
[39] J.-L. Seng, H.-F. Yang, The association between stock price volatility and financial news–a sentiment analysis approach, Kybernetes 46 (8) (2017) 1341–1365.
[40] L. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning, Part I Inf. Sci. 8 (1975) 199–249, Part II, Information Sciences 8 (1975) 301-357, Part III, Information Sciences 9 (1975) 43-80.
[41] F. Herrera, L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words, IEEE Trans. Fuzzy Syst. 8 (6) (2000) 746–752.
[42] F. Herrera, E. Herrera-Viedma, Aggregation operators for linguistic weighted information, IEEE Trans. Syst. Man Cybern. A 27 (1997) 646–656.
[43] F. Herrera, L. Martínez, A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic contexts in multiexpert decision-making, IEEE Trans. Syst. Man Cybern. B 31 (2) (2001) 227–234.
[44] L. Zhang, S. Wang, B. Liu, Deep learning for sentiment analysis: A survey, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 8 (4) (2018) e1253.
[45] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, 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, 2013, pp. 1631–1642.
[46] Z. Jianqiang, G. Xiaolin, Z. Xuejun, Deep convolution neural networks for twitter sentiment analysis, IEEE Access 6 (2018) 23253–23260.
[47] M. Sahlgren, The distributional hypothesis, Ital. J. Disabil. Stud. 20 (2008) 33–53.
[48] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, 2014, pp. 1188–1196.
[49] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space. Cornell university library. 2013, 2016, arXiv preprint arXiv:1301.3781.
[50] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, T. Mikolov, Fasttext.zip: Compressing text classification models, 2016, arXiv preprint arXiv:1612.03651.
[51] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[52] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[53] R. Johnson, T. Zhang, Semi-supervised convolutional neural networks for text categorization via region embedding, in: Advances in Neural Information Processing Systems, 2015, pp. 919–927.
[54] X. Wang, Y. Liu, S. Chengjie, B. Wang, X. Wang, Predicting polarities of tweets by composing word embeddings with long short-term memory, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Vol. 1, 2015, pp. 1343–1353.
[55] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, B. Qin, Learning sentiment-specific word embedding for twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2014, pp. 1555–1565.
[56] S. Merello, A.P. Ratto, Y. Ma, L. Oneto, E. Cambria, Investigating timing and impact of news on the stock market, in: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2018, pp. 1348–1354.
[57] R. Dridan, S. Oepen, Tokenization: Returning to a Long Solved Problem—A Survey, Contrastive Experiment, Recommendations, and Toolkit—, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vol. 2, 2012, pp. 378–382.
[58] A. Schofield, M. Magnusson, D. Mimno, Pulling out the stops: Rethinking stopword removal for topic models, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, 2017, pp. 432–436.
[59] H. Schmid, Improvements in part-of-speech tagging with an application to german, in: In Proceedings of the Acl Sigdat-Workshop, Citeseer, 1995.
[60] K. Chen, Z. Zhang, J. Long, H. Zhang, Turning from tf-idf to tf-igm for term weighting in text classification, Expert Syst. Appl. 66 (2016) 245–260.
[61] R.W. Masulis, L. Shivakumar, Does market structure affect the immediacy of stock price responses to news?, J. Financ. Quant. Anal. 37 (4) (2002) 617–648.

Please cite this article as: J. Bernabé-Moreno, A. Tejeda-Lorente, J. Herce-Zelaya et al., A context-aware embeddings supported method to extract a fuzzy sentiment polarity dictionary, Knowledge-Based Systems (2019) 105236, https://doi.org/10.1016/j.knosys.2019.105236.