A Supervised Multi-class Multi-label Word Embeddings Approach for Toxic Comment Classification

Salvatore Carta, Andrea Corriga, Riccardo Mulas, Diego Reforgiato Recupero and Roberto Saia
Department of Mathematics and Computer Science, University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy

Keywords: Apache Spark, Word Embeddings, Sentiment Analysis, Supervised Approach.

Abstract: Nowadays, communications made by using the modern Internet-based opportunities have revolutionized the way people exchange information, allowing real-time discussions among a huge number of users. However, the advantages offered by such powerful instruments of communication are sometimes jeopardized by the dangers related to personal attacks that lead many people to leave a discussion that they were participating. Such a problem is related to the so-called toxic comments, i.e., personal attacks, verbal bullying and, more generally, an aggressive way in which many people participate in a discussion, which brings some participants to abandon it. By exploiting the Apache Spark big data framework and several word embeddings, this paper presents an approach able to operate a multi-class multi-label classification of a discussion within a range of six classes of toxicity. We evaluate such an approach by classifying a dataset of comments taken from the Wikipedia's talk page, according to a Kaggle challenge. The experimental results prove that, through the adoption of different sets of word embeddings, our supervised approach outperforms the state-of-the-art that operate by exploiting the canonical bag-of-word model. In addition, the adoption of a word embeddings defined in a similar scenario (i.e., discussions related to e-learning videos), proves that it is possible to improve the performance with respect to solutions employing state-of-the-art word embeddings.

1 INTRODUCTION

The on-line communications between people generate a huge amount of data, giving life to what the researchers defined *Big Data*: a very huge dataset of information from which it is difficult to extract useful information in a reasonable time, if we do not use specific tools and strategies (e.g., *Machine Learning*, *Natural Language Processing*, etc.). Some examples of such sources of data are the communications related to *social networks*, *blogs*, and so on, an ever-increasing number of information that many companies exploit in order to offer free or paid services to their users (e.g., targeted recommendation of products and services).

In this scenario the downside is represented by the risks associated with toxic comments, since the advantages related to the aforementioned kind of information (social networks comments, reviews, politic opinions, etc.) are dramatically reduced by those who intervene with verbal attacks, verbal bullying, threats to the person, harassment and, more generally, behaviors that lead some participants to abandon the discussion.

Considering that an automatic approach designed to classify the toxicity related to a text must face a multi-class multi-label problem, because a text can be classified in more than a class, in order to define an effective, flexible, and scalable approach able to tackle this problem, this paper proposes a novel method that exploits the following three components:

1. the first component is the Apache Spark, which is used as big data framework with its Machine Learning library (MLlib);
2. the second component is the semantics of words obtained by using the word embeddings representations, a modeling approach that can be profitably applied in different domains (Boratto et al., 2016b; Boratto et al., 2016a);
3. the last component is a huge number of reviews taken from Udemy[1], which represents an e-learning platform with more than 65000 video courses.

Our strategy is to use several combinations of word embeddings obtained by exploiting different

[1] http://www.udemy.com

Carta, S., Corriga, A., Mulas, R., Recupero, D. and Saia, R.
A Supervised Multi-class Multi-label Word Embeddings Approach for Toxic Comment Classification.
DOI: 10.5220/0008110901050112
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 105-112
ISBN: 978-989-758-382-7

tools. For this operation we have used as data source both the Udemy dataset collection and other standard collections. Our goal is to compare the performance of our approach that uses standard word embeddings generated from news sources with the same approach that uses word embeddings generated from user comments to e-learning courses. The obtained results have confirmed our hypothesis, since it is possible to improve the classification performance by using word embeddings from a domain closer to that taken into account, outperforming the baselines solutions.

Hence, the main contributions of this paper to the state of the art are the following:

- we propose a scalable and flexible approach based on Apache Spark and the MLlib library for toxicity detection;
- we exploit different combinations of word embeddings to evaluate the performance of our approach;
- we use a huge dataset collected in a past work (Dessì et al., 2018) based on Udemy reviews;
- we define a multi-class multi-label classification approach that exploits the word embeddings and it is able to outperform the state-of-the-art solutions in terms of accuracy; to note that we have turned the multi-label multi-class problem in six different binary classification problems;
- we demonstrate that the approach employing word embeddings generated out of Udemy outperforms the approach using state-of-the-art word embeddings;
- we face a Kaggle[2] task, obtaining competitive results;
- we provide our source code publicly through GitHub[3].

2 RELATED WORKS

In this paper we face a Sentiment Analysis problem, since we deal with a research domain aimed to perform polarity detection (Devitt and Ahmad, 2007) in a given text. The problem has been defined within a Kaggle challenge where different researchers and people from industry participated. Recently, several other challenges within the Sentiment Analysis domain have been defined as well (Buscaldi et al., 2018; Recupero et al., 2017; Dragoni and Recupero, 2016; Recupero et al., 2015a; Recupero and Cambria, 2014). Generally, Sentiment Analysis polarity detection can be performed at two levels: binary level or fine-grained level. Whereas the former deals with assigning a positive/negative class to a certain text document, in the latter we can perform a multi-class classification or a regression (i.e., we classify a text in a range of values). Moreover, other tasks within the Sentiment Analysis domain involve the figurative-language detection (Filatova, 2012), and the aspect-based Sentiment Analysis (Federici and Dragoni, 2016).

2.1 Polarity Detection

Literature offers a number of *supervised*, *unsupervised*, or *hybrid* approaches able to perform polarity detection tasks. The *Supervised approaches* use labeled data in order to train sentiment classifiers, operating such as bag-of-words (Dridi and Reforgiato Recupero, 2017), micro-blogging features (Agarwal et al., 2011), *n*-grams with part-of-speech (POS) tags (Go et al., 2009), hashing features (da Silva et al., 2014), and so on. Considering that the effectiveness of such *supervised* approaches is reduced by the domain dependency on annotated training data, the *unsupervised approaches* are lexicon-based and they use pre-built lexicons of words weighted with their sentiment orientations in order to classify the overall sentiment related to a text (Momtazi, 2012). It should be observed that, given the huge amount of labeled data, *unsupervised approaches* are not able to outperform the *supervised* approach. *Hybrid* approaches that combined both methods (Maas et al., 2011) have been proposed as well.

2.2 Natural Language Processing

By exploiting the advantages related to semantics, a number of approaches able to extract sentiment and opinion by employing *Natural Language Processing* (NLP) techniques has been developed according to the regular and irregular, explicit and implicit, syntactical and semantic rules that regulate a specific language (Saia et al., 2016). In this research area, Cambria et al. (Cambria et al., 2012) have publicly provided *SenticNet*, a resource for Opinion Mining made by exploiting techniques based on the artificial intelligence and Semantic Web in order to perform an accurate and multi-faceted analysis of the natural language in a given text.

In addition, SenticNet has been used with Con-

[2] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[3] https://github.com/riccardomulas/Toxic-Comment-Classification

Table 1: Training and test sets: type of toxicity occurrences and percentage.

Toxicity	Training set occurrences	% with respect to the whole training set (159,571)	Test set occurrences	% with respect to the whole test set (63,978)
toxic	15,294	9,58%	6,090	9,51%
severe_toxic	1,595	0,99%	367	0,57%
obscene	8,449	5,29%	3,691	5,76%
threat	478	0,29%	211	0,32%
insult	7,877	4,93%	3,427	5,35%
identity_hate	1,405	0,88%	712	1,11%

ceptNet (Liu and Singh, 2004) in order to define an opinion-mining engine (Raina, 2013) able to perform a fine-grained Sentiment Analysis aimed to classify sentences (as positive, negative or neutral) from news articles. In this research area, Reforgiato Recupero et al. (Recupero et al., 2015b; Gangemi et al., 2014; Recupero et al., 2014) defined *Sentilo*, a sentic computing system for Sentiment Analysis able to combine the natural language processing techniques with the knowledge representation, exploiting affective knowledge resources such as SenticNet (Cambria et al., 2010), SentiWordNet (Baccianella et al., 2010), and SentiloNet (Recupero et al., 2015b).

2.3 Toxic Comment Classification

The objective of the proposed work is more specific than a general emotion detection task, since we want to analyze and evaluate the toxicity present in a given text, detecting the presence of some kind of expressions and the associate level of toxicity. In this scenario the concept of toxicity is related to concept of verbal violence made by operating personal attacks, on-line harassment, bullying behaviors and, in general, any disrespectful comment that can lead people to abandon an on-line conversation.

These are not isolated cases, because a recent Pew Report[4] claims that four in ten Americans have suffered such an on-line harassment, and many of them (62%) consider this a serious problem. It should be noted that the last report indicated that many of these people asked for new technologies able to face this problem, although they are in doubt about how to balance the free speech and the on-line safety.

Nowadays, this kind of problem is faced through a number of approaches and strategies based on machine learning or deep learning technologies (Parekh and Patel, 2017). For instance, in (Georgakopoulos et al., 2018) Convolutional Neural Network (CNN) for toxic comments detection have been proposed, comparing their effectiveness to the canonical approaches based on the bag-of-words models. The obtained results demonstrate that CNN is able to improve the toxic comment classification. Other studies have employed a supervised learning approach for this task (Yin et al., 2009) or a framework able to detect negative on-line interactions through text messages or images (Kansara and Shekokar, 2012). The study presented in (Warner and Hirschberg, 2012) has taken into account the hate text oriented towards specific group characteristics, such as ethnic origin, religion, gender, or sexual orientation, proving that such actions are characterized by the use of a small set of high frequency stereotypical words.

3 ADOPTED DATASET

The dataset we adopted in order to evaluate the proposed approach is the same used during the Toxic Comment Classification Challenge launched by Kaggle. Kaggle represents a well-known platform that proposes numerous predictive modeling and analytics competitions where the participants have to propose the best prediction model for a given dataset and for a certain task. As far as the Toxic Comment Classification Challenge is concerned, the related dataset has been provided by the Conversation AI team[5], a team founded by Jigsaw and Google. Such a dataset contains the comments from Wikipedia's talk page, which have been labeled by human raters for toxic behavior.

In more detail, they have identified six types of toxicity: *toxic*, *severe_toxic*, *obscene*, *threat*, *insult*, and *identity_hate*. Each of them presents a different grade of toxicity and a text is classified:

- as *toxic* when it contains strong expressions such as *"Don't look, come or think of coming back! Tosser."*;
- as *severe_toxic* when contains insults, swear words, etc.;
- as *Obscene* in presence of depravity;
- as *threat* when it contains threats such as *"Please stop. If you continue to ignore our policies by introducing inappropriate pages to Wikipedia, you will be blocked"*;

[4] https://pewrsr.ch/2u9X4aC

[5] https://conversationai.github.io/

- as *insult* if it contains insults like *"You are clearly not very smart and not here to build an encyclopedia"*;
- as *identity_hate* in presence of racial expressions.

Table 1 shows information about the aforementioned dataset and illustrates the details of the training and the test sets as defined by the Kaggle competition.

3.1 Word Embeddings

About the exploitation of the word embeddings, we adopted several and different combination of data that we have defined by applying state-of-the-art approaches for word model representation on the COCO dataset (Dessì et al., 2018). Such a dataset contains information gathered from Udemy, which represents one of the most important marketplaces for on-line learning. In more detail, it contains 1,2M user comments related to over 43,000 on-line courses at scale. A detailed description of the procedure adopted to gather the information is reported in (Dessì et al., 2018), whereas the details of the generated word embeddings is provided in Section 4.

4 PROPOSED APPROACH

We exploited the word embeddings information representation in order to extract meaningful features from the text, since this method allows us quantifying and categorizing the semantic similarities between linguistic items, on the basis of their distributional properties in large samples of language data. For this reason, our supervised approach of classification uses different combination of word embeddings. We evaluate such an approach in Section 5, where the advantages related to the word embeddings adoption have been underlined by the better results with respect to a baseline approach of classification based on the canonical bag-of-words strategy and the Term Frequency Inverse Document Frequency (TF-IDF) model. The five state-of-the-art word embeddings used during the experiments are reported in the following:

1. GloVe[6]: 6B tokens, 400K vocab, uncased, 100d and 300d vectors;
2. Google News dataset[7]: 300d vectors related tor 3 million of words and phrases;
3. SNAP Amazon dataset[8]: 135M product reviews on 27 categories whose embeddings have been created using the W2V algorithm with the deeplearning4j library[9];
4. FastText[10]: 1 million word vectors trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (16B tokens);
5. Dranziera (Dragoni et al., 2016): 1 million of reviews crawled from product pages on the Amazon web site that belong to twenty different categories. Vectors are of size 128, 256 and 512. We used the word embeddings created with 15 train epochs.

We have also defined further word embeddings based on the Udemy reviews (Dessì et al., 2018) by exploiting the FastText (Joulin et al., 2016), GloVe (Pennington et al., 2014), Word2Vec (Mikolov et al., 2013), and Intel (Ji et al., 2016) on-line tools. In addition, considering that a word embeddings data representation generates a very huge file, in order to improve their effectiveness and to reduce the computational load, we operated a data reduction by excluding all the vectors related to words that are not present in the current data.

As far as the *Apache Spark* (Zaharia et al., 2010) framework is concerned, it represents an open-source cluster computing framework able to provide high-level APIs in *Java*, *Scala*, *Python*, and *R* languages, plus an engine optimized for general execution graphs. It also supports a rich set of higher-level tools such as *MLlib* for machine learning, *SparkSQL* for SQL and structured data processing, *GraphX* for graph processing, and *Spark Streaming* for streaming analytics. Apache Spark has been mainly chosen for its performance in terms of computation time and scalability. We have also exploited its scalable library MLlib for machine learning. In order to overcome the lack of compatibility that characterizes Apache Spark respect to the SQL DataFrames (Armbrust et al., 2015), when it needs to deal with high dimension files, before the classification, we defined two UDF (User-Defined Function) able to combine the word embeddings with the tokenized words of the dataset, converting the resulting vector (NumPy array) in a native Python format (VectorUDT) compatible with the model.

4.1 Evaluation Metrics

The metrics we have chosen in order to evaluate the experimental results are the *accuracy* and the *AUC*

[6] https://nlp.stanford.edu/projects/glove/

[7] https://bit.ly/1VxNC9t

[8] https://snap.stanford.edu/data/web-Amazon.html

[9] https://deeplearning4j.org/word2vec.html

[10] https://fasttext.cc/docs/en/english-vectors.html

Table 2: Accuracy and AUC per label for each state-of-the-art word embeddings (SET$_1$).

Label	G. News	GloVe 100d	GloVe 300d	FastText	SNAP	Dranz 128d	Dranz 256d	Dranz 512d
toxic	0.90/0.77	0.88/0.76	0.90/0.80	0.91/0.85	0.89/0.82	0.87/0.8	0.89/0.82	0.91/0.81
severe_toxic	0.91/0.82	0.96/0.83	0.97/0.83	0.96/0.89	0.91/0.84	0.9/0.85	0.92/0.82	0.93/0.84
obscene	0.94/0.77	0.93/0.75	0.94/0.77	0.95/0.82	0.92/0.81	0.87/0.78	0.9/0.81	0.91/0.8
threat	0.93/0.75	0.94/0.74	0.95/0.75	0.96/0.81	0.92/0.80	0.88/0.77	0.92/0.79	0.93/0.8
insult	0.95/0.76	0.93/0.76	0.93/0.77	0.92/0.82	0.95/0.80	0.86/0.74	0.9/0.76	0.91/0.77
identity_hate	0.94/0.73	0.96/0.72	0.97/0.72	0.93/0.75	0.9/0.78	0.86/0.71	0.89/0.72	0.9/0.74
Average	0.93/0.77	0.93/0.76	0.94/0.77	0.94/0.82	0.92/0.81	0.87/0.77	0.9/0.79	0.92/0.79

Table 3: Accuracy and AUC per label for each state-of-the-art word embeddings (SET$_2$).

Label	G. News	GloVe 100d	GloVe 300d	FastText	SNAP	Dranz 128d	Dranz 256d	Dranz 512d
toxic	0.88/0.75	0.87/0.74	0.87/0.77	0.9/0.84	0.86/0.81	0.85/0.78	0.86/0.75	0.88/0.77
severe_toxic	0.88/0.8	0.95/0.82	0.94/0.81	0.94/0.85	0.88/0.8	0.87/0.76	0.9/0.77	0.91/0.78
obscene	0.92/0.74	0.92/0.73	0.92/0.72	0.9/0.8	0.89/0.78	0.84/0.78	0.86/0.75	0.87/0.76
threat	0.9/0.72	0.91/0.71	0.92/0.73	0.93/0.79	0.9/0.76	0.85/0.77	0.91/0.75	0.91/0.76
insult	0.93/0.74	0.92/0.72	0.91/0.73	0.89/0.78	0.93/0.77	0.84/0.78	0.88/0.76	0.9/0.79
identity_hate	0.92/0.72	0.93/0.71	0.95/0.71	0.92/0.72	0.89/0.75	0.85/0.78	0.87/0.8	0.89/0.79
Average	0.9/0.75	0.92/0.74	0.92/0.75	0.91/0.8	0.89/0.78	0.85/0.77	0.88/0.76	0.89/0.77

(Area Under the ROC[11] Curve), as detailed in the following.

4.1.1 Accuracy

The first one (i.e. accuracy) is a metric based on the *confusion matrix*, according with the formalization shown in Equation 1, where tp, tn, fp, and fn are, respectively, true positives, true negatives, false positives, and false negatives.

$$accuracy = \frac{tp+tn}{tp+tn+fp+fn} \qquad (1)$$

4.1.2 AUC

The second used metric (i.e., AUC) is formalized in Equation 2, where given the subsets X and Y, α indicates all the possible comparisons between these subsets and the result is obtained by averaging all the comparisons and lies within the interval $[0,1]$, where *1* denotes the best performance. It is a metric largely used in order to evaluate the performance of a classification model and, in addition, it is the same metric used during by the Kaggle competition where the toxicity classification task has been proposed.

$$\alpha(X,Y) = \begin{cases} 1, & if\, x > y \\ 0.5, & if\, x = y \\ 0, & if\, x < y \end{cases} \quad AUC = \frac{1}{|X| \cdot |Y|} \sum_{1}^{|X|} \sum_{1}^{|Y|} \alpha(x,y) \qquad (2)$$

5 EXPERIMENTS

We performed two sets of experiments with the aim of comparing several word embeddings approaches, i.e., those based on the state-of-the-art solutions and those built using Udemy data. Moreover, we compared approaches employing word embeddings against the baseline that does not use word embeddings.

In the first set (SET_1), a logistic regression classifier is trained by using the training set and tested by using the test set, whereas in the second set (SET_2), we merged the training and test sets and adopted a k-cross validation criterion with $k = 10$ to avoid potential biases present in the assigned training and test sets.

Table 6 shows accuracy and AUC results related to the two baseline approaches using a logistic regression classifier without word embeddings and with BOW model with either term frequency or TF-IDF representation (using unigrams) for SET_1 and SET_2. Such metrics underline how the accuracy and the AUC related to the model without the usage of word embeddings is better when the TF-IDF distance is employed with respect to the TF for both SET_1 and SET_2.

Table 2 indicates the accuracy and AUC values for each used state-of-the-art word embeddings for SET_1 (see Section 3), whereas Table 3 shows similar results for SET_2.

Finally, Table 4 and Table 5 report the accuracy and AUC measured for each word embeddings generated out of the reviews collection extracted from Udemy with 50 train epochs for SET_1 and SET_2. It should be observed that in all the tables the values related to the two metrics taken into account are presented in the form accuracy/AUC.

As shown in Tables 2, 3, 4 and 5, for each word embeddings and for both SET_1 and SET_2, the results obtained on each of the six toxic classes and their average are better when employing word embeddings. Moreover, Tables 4 and 5 show that results obtained by using contextual word embeddings (Udemy) outperform those general-purpose (texts that belong to several contexts) word embeddings.

Experimental results show how the accuracy and AUC measured in SET_1 are always higher than those measured in SET_2, indicating that the Kaggle test

[11] Receiver Operating Characteristic

Table 4: Accuracy and AUC per label for each Udemy reviews word embeddings with dimensions 100 or 300, both with 50 train epochs (SET_1).

Label	$FastText_{100}$	$FastText_{300}$	$GloVe_{100}$	$GloVe_{300}$	$Word2Vec_{100}$	$Word2Vec_{300}$	$Intel_{100}$	$Intel_{300}$
toxic	0.94/0.83	0.95/0.87	0.92/0.80	0.93/0.82	0.92/0.84	0.93/0.85	0.95/0.84	0.95/0.85
severe_toxic	0.95/0.91	0.95/0.92	0.93/0.89	0.96/0.89	0.96/0.9	0.96/0.9	0.96/0.89	0.96/0.93
obscene	0.96/0.85	0.96/0.88	0.94/0.84	0.96/0.82	0.96/0.84	0.95/0.87	0.95/0.84	0.95/0.83
threat	0.96/0.81	0.96/0.82	0.97/0.83	0.97/0.84	0.93/0.85	0.96/0.87	0.96/0.86	0.96/0.88
insult	0.95/0.87	0.96/0.88	0.96/0.89	0.96/0.9	0.95/0.88	0.96/0.87	0.97/0.89	0.96/0.89
identity_hate	0.97/0.89	0.96/0.88	0.96/0.88	0.96/0.88	0.94/0.88	0.96/0.89	0.97/0.89	0.97/0.88
Average	0.96/0.86	0.96/0.88	0.95/0.86	0.96/0.86	0.94/0.87	0.95/0.88	0.96/0.87	0.96/0.88

Table 5: Accuracy and AUC per label for each Udemy reviews word embeddings with dimensions 100 or 300, both with 50 train epochs (SET_2).

Label	$FastText_{100}$	$FastText_{300}$	$GloVe_{100}$	$GloVe_{300}$	$Word2Vec_{100}$	$Word2Vec_{300}$	$Intel_{100}$	$Intel_{300}$
toxic	0.93/0.9	0.92/0.86	0.9/0.79	0.91/0.81	0.92/0.81	0.93/0.82	0.92/0.8	0.93/0.85
severe_toxic	0.94/0.86	0.94/0.89	0.91/0.85	0.93/0.86	0.95/0.87	0.94/0.86	0.93/0.87	0.95/0.89
obscene	0.93/0.82	0.94/0.83	0.91/0.8	0.95/0.79	0.91/0.8	0.92/0.84	0.94/0.8	0.93/0.83
threat	0.94/0.78	0.94/0.81	0.94/0.8	0.96/0.82	0.92/0.83	0.93/0.84	0.95/0.85	0.95/0.85
insult	0.93/0.83	0.93/0.83	0.94/0.82	0.95/0.83	0.94/0.82	0.93/0.82	0.94/0.83	0.95/0.83
identity_hate	0.92/0.82	0.93/0.83	0.94/0.83	0.93/0.83	0.92/0.82	0.94/0.81	0.95/0.83	0.95/0.83
Average	0.93/0.84	0.93/0.84	0.92/0.82	0.94/0.82	0.93/0.83	0.93/0.83	0.94/0.83	0.95/0.85

Table 6: Accuracy per label for baseline using term frequency (TF) and TF-IDF.

Label	Accuracy (TF/TF-IDF) SET_1	AUC (TF/TF-IDF) SET_1	Accuracy (TF/TF-IDF) SET_2	AUC (TF/TF-IDF) SET_2
toxic	0.81/0.85	0.76/0.73	0.8/0.83	0.74/0.72
severe_toxic	0.85/0.91	0.77/0.76	0.84/0.91	0.73/0.74
obscene	0.84/0.88	0.67/0.74	0.84/0.87	0.66/0.75
threat	0.82/0.90	0.75/0.74	0.81/0.88	0.72/0.75
insult	0.80/0.89	0.74/0.75	0.78/0.88	0.73/0.72
identity_hate	0.84/0.91	0.72/0.75	0.83/0.9	0.7/0.74
Average	0.83/0.89	0.74/0.75	0.82/0.88	0.71/0.74

set represents a fold less important than the others for what the training phase is concerned. That is, the test set contains elements that are not distinctive and whose behaviour is similar to that of the other folds. Therefore, in the context of the adopted k-fold-validation process, the test set is mixed with the other data and, as consequence, the average accuracy and AUC resulting from each step is decreased.

6 CONCLUSION AND FUTURE WORK

Nowadays, the toxic comment classification task represents a problem more and more important, since it jeopardizes the big opportunities offered by the online instruments of communication such as social networks, blogs, forums, and so on. This paper proposes a machine learning approach aimed at facing this problem through the adoption of the Apache Spark framework an its capability to deal with Big Data by using the Machine Learning library (MlLib).

During the performed experiments we made two types of comparisons. The first one by using the same classification approach with and without the state-of-the-art word embeddings data representation, demonstrating that the word embeddings are able to improve the classification performance with regard to the baseline methods using bag of words models. The second one by defining word embeddings out of users' comments related to Udemy e-learning platform, with the hypothesis that the creation of word embeddings made by using data closer to the domain taken into account should improve the classification performance. Such an hypothesis has been confirmed by the performed experiments in all the different methods used to define the word embeddings, because the approach employing Udemy embeddings outperforms those using the state-of-the-art word embeddings.

Summarizing, the obtained results indicate that the adoption of word embeddings are able to improve the accuracy with regard to the baselines approaches that do not adopt word embeddings. In addition, such results prove that the contextual word embeddings outperform the canonical state-of-the-art word embeddings in a specific domain. An interesting future work would be the experimentation of a combination of different contextual word embeddings and deep learning approaches, as well as an ensemble strategy, in order to improve the overall performance.

ACKNOWLEDGEMENTS

This research is partially funded by Italian Ministry of Education, University and Research - Program Smart Cities and Communities and Social Innovation project ILEARNTV (D.D. n.1937 del 05.06.2014, CUP F74G14000200008 F19G14000910008). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Armbrust, M., Xin, R. S., Lian, C., Huai, Y., Liu, D., Bradley, J. K., Meng, X., Kaftan, T., Franklin, M. J., Ghodsi, A., and Zaharia, M. (2015). Spark sql: Relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 1383–1394, New York, NY, USA. ACM.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 2200–2204, Valletta, Malta. European Language Resources Association (ELRA).

Boratto, L., Carta, S., Fenu, G., and Saia, R. (2016a). Representing items as word-embedding vectors and generating recommendations by measuring their linear independence. *RecSys Posters*, 140.

Boratto, L., Carta, S., Fenu, G., and Saia, R. (2016b). Using neural word embeddings to model user behavior and detect user segments. *Knowledge-Based Systems*, 108:5–14.

Buscaldi, D., Gangemi, A., and Recupero, D. R., editors (2018). *Semantic Web Challenges - 5th SemWebEval Challenge at ESWC 2018, Heraklion, Greece, June 3-7, 2018, Revised Selected Papers*, volume 927 of *Communications in Computer and Information Science*. Springer.

Cambria, E., Havasi, C., and Hussain, A. (2012). SenticNet 2: A Semantic and Affective Resource for Opinion Mining and Sentiment Analysis. In Youngblood, G. M. and McCarthy, P. M., editors, *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, pages 202–207. AAAI Press.

Cambria, E., Speer, R., Havasi, C., and Hussain, A. (2010). SenticNet: A Publicly Available Semantic Resource for Opinion Mining. In *AAAI Fall Symposium: Commonsense Knowledge*, volume FS-10-02 of *AAAI Technical Report*, pages 14–18. AAAI.

da Silva, N. F., Hruschka, E. R., and Hruschka, E. R. (2014). Tweet Sentiment Analysis with Classifier Ensembles. *Decis. Support Syst.*, 66(C):170–179.

Dessì, D., Fenu, G., Marras, M., and Reforgiato Recupero, D. (2018). Coco: Semantic-enriched collection of online courses at scale with experimental use cases. In Rocha, Á., Adeli, H., Reis, L. P., and Costanzo, S., editors, *Trends and Advances in Information Systems and Technologies*, pages 1386–1396. Springer International Publishing.

Devitt, A. and Ahmad, K. (2007). Sentiment polarity identification in financial news: A cohesion-based approach. pages 984–991, Prague, CZ. Association for Computational Linguistics.

Dragoni, M. and Recupero, D. R. (2016). Challenge on fine-grained sentiment analysis within ESWC2016. In *Semantic Web Challenges - Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, pages 79–94.

Dragoni, M., Tettamanzi, A. G., and Pereira, C. D. C. (2016). Dranziera: an evaluation protocol for multi-domain opinion mining. In *Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 267–272. European Language Resources Association (ELRA).

Dridi, A. and Reforgiato Recupero, D. (2017). Leveraging semantics for sentiment polarity detection in social media. *International Journal of Machine Learning and Cybernetics*.

Federici, M. and Dragoni, M. (2016). A knowledge-based approach for aspect-based opinion mining. In Sack, H., Dietze, S., Tordai, A., and Lange, C., editors, *Semantic Web Challenges*, pages 141–152, Cham. Springer International Publishing.

Filatova, E. (2012). Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Proceedings of LREC 2012*, pages 392–398.

Gangemi, A., Presutti, V., and Recupero, D. R. (2014). Frame-based detection of opinion holders and topics: A model and a tool. *IEEE Comp. Int. Mag.*, 9(1):20–30.

Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., and Plagianakos, V. P. (2018). Convolutional neural networks for toxic comment classification. *CoRR*, abs/1802.09957.

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

Ji, S., Satish, N., Li, S., and Dubey, P. (2016). Parallelizing word2vec in shared and distributed memory. *CoRR*, abs/1604.04661.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. *arXiv:1612.03651*.

Kansara, K. B. and Shekokar, N. M. (2012). A framework for cyberbullying detection in social network. volume 5, pages 494–498.

Liu, H. and Singh, P. (2004). ConceptNet – A Practical Commonsense Reasoning Tool-Kit. *BT Technology Journal*, 22(4):211–226.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Momtazi, S. (2012). Fine-grained German Sentiment Analysis on Social Media. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 1215–1220.

Parekh, P. and Patel, H. (2017). Toxic comment tools: A case study. *International Journal of Advanced Research in Computer Science*, 8(5):964–967.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Raina, P. (2013). Sentiment Analysis in News Articles Using Sentic Computing. In *Proceedings of the 2013 IEEE 13th International Conference on Data Mining Workshops*, ICDMW '13, pages 959–962, Washington, DC, USA. IEEE Computer Society.

Recupero, D. R. and Cambria, E. (2014). Eswc'14 challenge on concept-level sentiment analysis. In *Semantic Web Evaluation Challenge - SemWebEval 2014 at ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, pages 3–20.

Recupero, D. R., Cambria, E., and Rosa, E. D. (2017). Semantic sentiment analysis challenge at ESWC2017. In *Semantic Web Challenges - 4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, May 28 - June 1, 2017, Revised Selected Papers*, pages 109–123.

Recupero, D. R., Consoli, S., Gangemi, A., Nuzzolese, A. G., and Spampinato, D. (2014). A semantic web based core engine to efficiently perform sentiment analysis. In *The Semantic Web: ESWC 2014 Satellite Events - ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, pages 245–248.

Recupero, D. R., Dragoni, M., and Presutti, V. (2015a). ESWC 15 challenge on concept-level sentiment analysis. In *Semantic Web Evaluation Challenges - Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*, pages 211–222.

Recupero, D. R., Presutti, V., Consoli, S., Gangemi, A., and Nuzzolese, A. G. (2015b). Sentilo: Frame-Based Sentiment Analysis. *Cognitive Computation*, 7(2):211–225.

Saia, R., Boratto, L., Carta, S., and Fenu, G. (2016). Binary sieves: Toward a semantic approach to user segmentation for behavioral targeting. *Future Generation Computer Systems*, 64:186–197.

Warner, W. and Hirschberg, J. (2012). Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, LSM '12, pages 19–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yin, D., Xue, Z., Davison, B. D., and Edwards, L. (2009). Detection of harassment on web 2 . 0. In *In Proceedings of the Content Analysis in the WEB 2.0 (CAW2.0) Workshop at WWW2009, Madrid, Spain, April 2009.*

Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: Cluster Computing with Working Sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10, Berkeley, CA, USA. USENIX Association.