Neural Computing and Applications (2020) 32:6793–6807
https://doi.org/10.1007/s00521-019-04334-2

BRAIN INSPIRED COMPUTING & MACHINE LEARNING APPLIED RESEARCH-BISMLARE

A deep learning classifier for sentence classification in biomedical and computer science abstracts

Sérgio Gonçalves[1] · Paulo Cortez[1] · Sérgio Moro[2]

Received: 27 December 2018 / Accepted: 28 June 2019 / Published online: 10 July 2019

Abstract

The automatic classification of abstract sentences into its main elements (background, objectives, methods, results, conclusions) is a key tool to support scientific database querying, to summarize relevant literature works and to assist in the writing of new abstracts. In this paper, we propose a novel deep learning approach based on a convolutional layer and a bidirectional gated recurrent unit to classify sentences of abstracts. First, the proposed neural network was tested on a publicly available repository containing 20 thousand abstracts from the biomedical domain. Competitive results were achieved, with weight-averaged Precision, Recall and F1-score values around 91%, and an area under the ROC curve (AUC) of 99%, which are higher when compared to a state-of-the-art neural network. Then, a crowdsourcing approach using gamification was adopted to create a new comprehensive set of 4111 classified sentences from the computer science domain, focused on social media abstracts. The results of applying the same deep learning modeling technique trained with 3287 (80%) of the available sentences were below the ones obtained for the larger biomedical dataset, with weight-averaged Precision, Recall and F1-score values between 73 and 76%, and an AUC of 91%. Considering the dataset dimension as a likely important factor for such performance decrease, a data augmentation approach was further applied. This involved the use of text mining to translate sentences of the computer science abstract corpus while retaining the same meaning. Such approach resulted in slight improvements (around 2 percentage points) for the weight-averaged Recall and F1-score values.

Keywords Bidirectional gated recurrent unit · Abstract sentence classification · Deep learning · Crowdsourcing

1 Introduction

The number of scholarly works has increased during in the last decades [1]. For example, around 114 million of English scholarly documents were accessible on the Web in 2014 [2]. Such volume makes it difficult to quickly select relevant scientific documents. Scientific abstracts summarize the most important elements of a paper, and thus, those are valuable sources for filtering the most relevant papers during a literature review process [3].

The classification of scientific abstracts is a particular instance of the sequential classification task, considering there is a typical order in the classes (e.g., the “Objective” label tends to appear after the “Background”) [4]. This classification transforms unstructured text into a more information manageable structure [5]. This is acknowledged by the Emerald publisher, which requires all submissions to include a structured abstract [6]. In effect, the automatic classification of abstract sentences presents several advantages. It is a valuable tool for general scientific database querying (e.g., using Web of Science, Scopus). Also, it can assist in manual [7] or text mining [8]

✉ Sérgio Moro
sergio.moro@iscte-iul.pt

Sérgio Gonçalves
a72886@alunos.uminho.pt

Paulo Cortez
pcortez@dsi.uminho.pt

[1] ALGORITMI Centre, Department of Information Systems, University of Minho, Guimarães, Portugal

[2] Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR-IUL, Lisboa, Portugal

systematic literature review processes, as well as other bibliometric analyses. Moreover, it can help in the writing of new paper abstracts [9].

In this study, we present a novel deep learning neural network architecture for the sequential classification of abstract sentences. The architecture uses a word embedding layer, a convolutional layer, a bidirectional gated recurrent unit (GNU) and a final concatenation layer. The proposed deep learning model is compared with a recently proposed bidirectional long short-term memory (LSTM)-based model [5], showing an interesting performance on classifying sentences under five classes: "Background," "Objectives," "Methods," "Results" and "Conclusions" in abstracts corpus from two distinct domains. First, the approach is evaluated using a large publicly available 20K biomedical abstract corpus [10], which was also previously studied using a different approach [5], thus enabling a direct comparison. As a secondary contribution, we build a purely new abstract corpus from the computer science domain, with a particular focus on the social media topic. The new corpus was created by adopting a crowdsourcing approach with gamification features to attract researchers to manually classify the sentences. By building on collective intelligence, more than 4k sentences were categorized. To further improve the classification results, we employed data augmentation based on text mining, thus doubling the training set size. The computer science abstract corpus was also made publicly available; thus, it can be used in future research comparison studies. We consider two distinct abstract sentence corpus, from different domains (biomedical and computer science), in order to provide a more robust validation. The good classification results that were obtained for both domains (Sect. 4) suggest that our approach is potentially valuable for sentence classification of abstracts from any scientific domain.

This paper is organized as follows: Sect. 2 presents the related work; Sect. 3 describes the two abstract corpus, the deep learning architecture and evaluation metrics; the obtained results are analyzed in Sect. 4; finally, conclusions are drawn in Sect. 5.

2 Related work

2.1 Deep learning for abstract sentence classification

As pointed out by Dernoncourt et al. [5], most sequential sentence classification methods are based on "shallow" methods (e.g., naive Bayes, support vector machines (SVM)) that require a manual feature engineering based on lexical (e.g., bag of words, n-grams), semantic (e.g., synonyms), structural (e.g., part-of-speech tags) or sequential (e.g., sentence position) information. The advantage of using deep learning is that the neural networks do not require such manual design of features. Also, deep learning often achieves competitive results in text classification [11].

Regarding abstract sentence classification, this topic has been scarcely researched when compared to other text classification tasks (e.g., sentiment analysis). The main reason for this reduced attention is the restricted availability of public datasets. In 2010 [12], the manual engineering approach was used to set nine features (e.g., bigrams) and train five classifiers (e.g., SVM) that were combined to classify four main elements of medical abstracts. In 2013 [9], a private corpus with 4550 abstracts from different scientific fields was collected from ScienceDirect. The abstract sentences were manually labeled into four categories: "Background," "Goal," "Methods" and "Results." The authors also used the conventional manual feature design approach (e.g., n-grams) and a transductive SVM. More recently, in 2017 [10], a large abstract corpus was made publicly available. Using this dataset, a deep learning model, based on one bidirectional LSTM, was proposed for a five class sentence prediction, outperforming four other approaches (e.g., n-gram logistic regression, multilayer perceptron) [5]. In this paper, we propose a different deep learning architecture, mainly composed by a convolutional layer and a bidirectional GRU layer to classify the sentences from abstracts, which uses word embeddings instead of character embeddings. By taking into consideration the position of the sentences, as well as encoding contextual information on the vector of each sentence, we expect that the proposed architecture can potentially achieve better results when compared to the study by Dernoncourt et al. [5].

2.2 Crowdsourcing and collective intelligence to classify data

The difficulties of gathering already classified datasets highlighted in the previous subsection raises the challenge of testing our approach in several domains. Yet, it is essential for a broader validation of the accuracy of our proposal. Furthermore, by building a new dataset and making it publicly available, it can constitute a baseline for future comparisons with distinct approaches of ours, helping to pave avenues for future research. Thus, this study sets the additional challenge of gathering a newly classified dataset from another domain to complement the biomedical dataset made publicly available by Dernoncourt and Lee [10].

Collective intelligence consists in benefiting from many individuals by combining their intelligence in cognitive tasks [13, 14]. Currently, the Internet has leveraged collective intelligence by bringing communities together online in discussion forums and groups in social networks. Thus, the technology currently available at the reach of a click instantaneously connects people who may be spread across the globe. As a result, the concept of crowdsourcing has emerged to take advantage from collective intelligence by using the Internet to gather people together toward a common goal of solving cognitive tasks [15], from data classification to solving complex mathematics problems. Specifically, Welinder and Perona [16] have shown that data annotation can be effectively done using a web platform available online offering a crowdsourcing service.

One of the main challenges of crowdsourcing is to keep participants motivated during the whole process. Thus, a crowdsourcing platform must include an incentive system to prompt for participation through positive encouragement [17]. Incentives and motivation are interconnected, since the former helps in building the latter [18]. Therefore, adopting strategies that keep individuals motivated is at the core of crowdsourcing [19]. The motivation may be extrinsic, through external incentives such as money or discount coupons, or intrinsic, when the individual is pleased just by participating on crowdsourcing [18]. While both extrinsic and intrinsic motives are shown to influence participation, some authors such as Zheng et al. [20] argue that the latter motives can be more effective than the former, although such difference may be also due to cultural factors. A possible solution freely available of intrinsic incentive is gamification. It offers users appealing awards granted on desired accomplishments to incentivize participation [21].

3 Materials and methods

3.1 Computer science abstract corpus retrieval and classification

The first task to build the comprehensive set of classified abstract sentences in computer science applied to social media is to retrieve the raw set of abstracts. Although there are several platforms which freely publish scientific abstracts, most of them do not show a well-defined policy about what can be publicly shown to third-parties and, specifically, in the form of a dataset. As such, we choose the arXiv, which allows using articles' metadata, including abstracts, and provides an API for both commercial and non-commercial usage.[1] To retrieve data, we adopted the *aRxiv R* package, which uses the API to efficiently search the archive. Within the "computer science" category, we searched only for articles containing the terms "social media," "social network" or "social networks" in the title.

The different human languages pose serious challenges in text mining and machine learning tasks and, as such, many researchers choose to focus in one language (e.g., [22]). Since most of scientific literature is currently published in the English language [23] and, additionally, the large publishers and top journals require submissions to be in this language, we filtered all the collected articles using the *textcat* package [24] to match English abstracts. The result is that from a total of 658, only 4 were written in other than the English language.

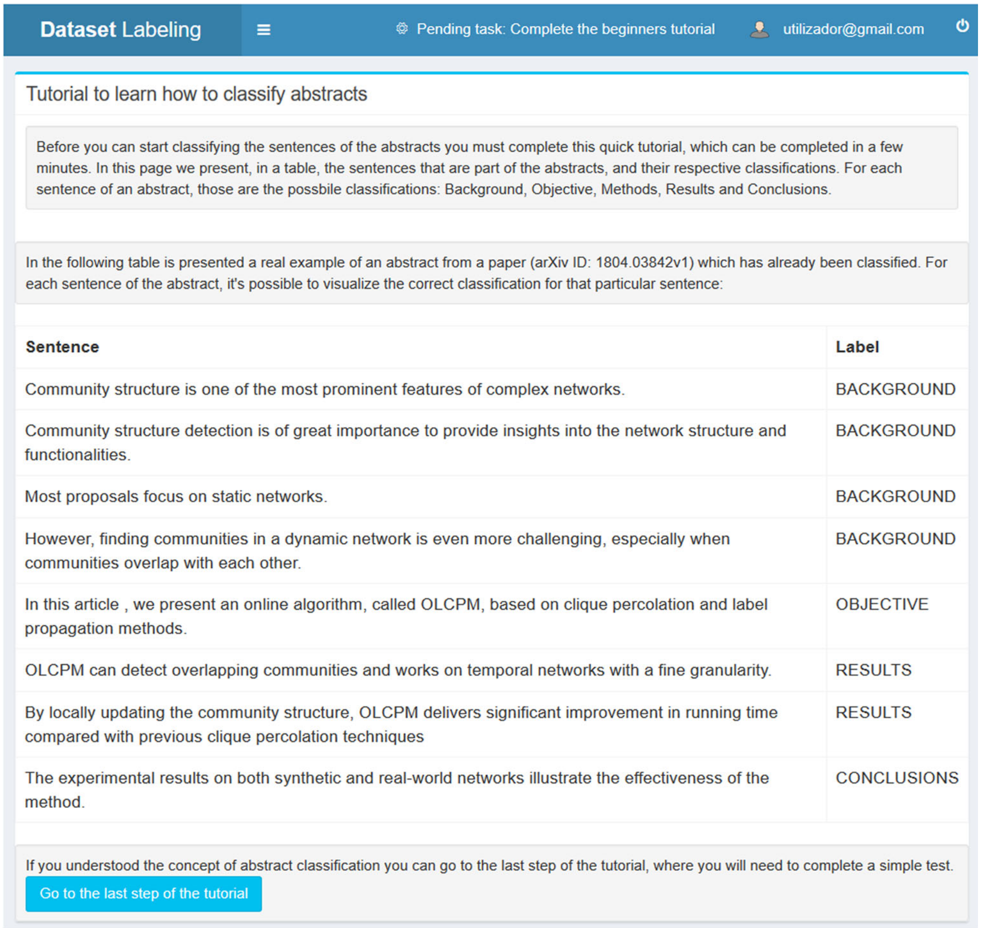
We implemented our crowdsourcing solution using the Amazon Elastic Compute Cloud, which is an Infrastructure-as-a-Service cloud platform. This is the platform with the largest market share, with around 40% of the Infrastructure-as-a-Service market [25], and it is freely available for academic purposes. The Ubuntu 16.04 operative system and the MariaDB database system were installed in the cloud, as well as the Nginx web server, which presents better performance and less memory requirements when compared to Apache, according to Suciu et al. [26]. We adopted the PHP Laravel (PHP 7.2) framework for developing our solution.

We made available our web responsive platform (adaptable depending on the type of device) in the following URL: http://classifyabstracts.info/. It was shared in academic social networks such as "www.researchgate.net," as well as through our own research contacts. The user needs to login to access an initial small tutorial that introduces the aims and scope of this research. This introductory tutorial consists in two steps. The first consists in reading a few examples of sentences previously classified, such that the user takes contact with the possible categories for the classification procedure (Fig. 1).

On the second and last step, the user needs to correctly classify each sentence of an abstract already classified. To simplify the annotation process, the categories for classification are selected through drop-down lists. If the user has at least one wrong classification, a warning appears highlighting the mistake and the user needs to repeat this step again. After successfully completing the tutorial, the user may start classifying unclassified abstracts in the platform (Fig. 2).

After the user classifies an abstract, a message is shown displaying the number of already classified abstracts. As an incentive, a simple gamification component consisting in points (one point per classified abstract) and three levels was introduced. Thus, a user with 10 points achieves the bronze level, while a user reaching 50 points achieves the silver level and, finally, a user with 100 or more points achieves the gold level.

[1] https://arxiv.org/help/oa/index.


Dataset Labeling
≡
Pending task: Complete the beginners tutorial
utilizador@gmail.com
Go to the last step of the tutorial

Tutorial to learn how to classify abstracts

Before you can start classifying the sentences of the abstracts you must complete this quick tutorial, which can be completed in a few minutes. In this page we present, in a table, the sentences that are part of the abstracts, and their respective classifications. For each sentence of an abstract, those are the possbile classifications: Background, Objective, Methods, Results and Conclusions.

In the following table is presented a real example of an abstract from a paper (arXiv ID: 1804.03842v1) which has already been classified. For each sentence of the abstract, it's possible to visualize the correct classification for that particular sentence:

Sentence	Label
Community structure is one of the most prominent features of complex networks.	BACKGROUND
Community structure detection is of great importance to provide insights into the network structure and functionalities.	BACKGROUND
Most proposals focus on static networks.	BACKGROUND
However, finding communities in a dynamic network is even more challenging, especially when communities overlap with each other.	BACKGROUND
In this article , we present an online algorithm, called OLCPM, based on clique percolation and label propagation methods.	OBJECTIVE
OLCPM can detect overlapping communities and works on temporal networks with a fine granularity.	RESULTS
By locally updating the community structure, OLCPM delivers significant improvement in running time compared with previous clique percolation techniques	RESULTS
The experimental results on both synthetic and real-world networks illustrate the effectiveness of the method.	CONCLUSIONS

If you understood the concept of abstract classification you can go to the last step of the tutorial, where you will need to complete a simple test.

Fig. 1 First step of the introductory tutorial

In total, 76 users classified abstracts, with very few of them classifying more than 100 abstracts, and still few reaching the silver level. Thus, most users devoted little effort and classified few abstracts. This is a known inequality phenomenon of crowdsourcing participants, in which only a small fraction of users holds a large contribution [27].

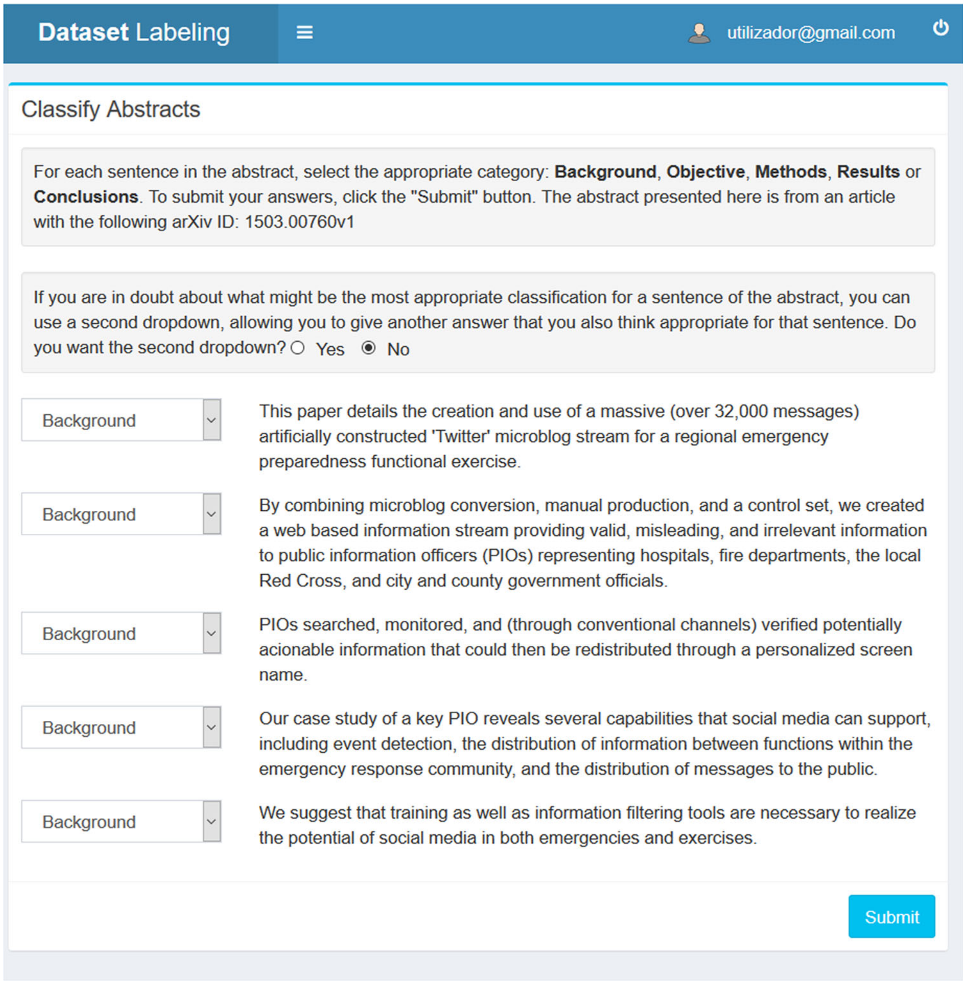
3.2 Scientific abstract corpora

Data understanding is a critical step in the knowledge discovery process [28]. Thus, in this subsection we characterize each of the two datasets used to test our approach. One of them is the one gathered through crowdsourcing (Sect. 3.1), while the other is the PubMed 20k dataset made publicly available by Dernoncourt and Lee [10]. The latter, from here forth referred to as “PubMed 20k,” sets the baseline for comparison purposes [29]. The former, labeled as “CS Abstracts,” enables to test our approach in a new domain and additionally provides to future researchers another corpus for sentence classification.

The corpus of the PubMed 20k includes open access papers from the PubMed biomedical database and related with randomized controlled trials (RCT). The sentences were classified by the authors of the articles into the five standardized labels. The full corpus has a total of 200K abstracts. A smaller subset, with 20K most recent abstracts, was also made available for a faster experimentation of sequential sentence classification methods. Considering the 20K subset was used in the work of Dernoncourt et al. [5], we also adopted the same dataset, to facilitate the experimental comparison.

The size of both datasets is substantially different, with the PubMed 20k containing a larger number of abstracts and, subsequently, more classified sentences. Table 1 shows the total number of abstracts and sentences used for training, validating the model, and testing it. The abstract distribution through the three partitions in the “CS Abstracts,” training, validation and testing, was done to evenly maintain distribution of each class. According to Ng [30], a uniform distribution between the three partitions in the datasets is used to prevent unexpected performance issues in productive environments. A model is trained using the training dataset, it is tuned according to the obtained results in the validation dataset, and its generalization capability is measured on the test dataset. As for the

Fig. 2 Second step of the introductory tutorial



Classify Abstracts

For each sentence in the abstract, select the appropriate category: **Background**, **Objective**, **Methods**, **Results** or **Conclusions**. To submit your answers, click the "Submit" button. The abstract presented here is from an article with the following arXiv ID: 1503.00760v1

If you are in doubt about what might be the most appropriate classification for a sentence of the abstract, you can use a second dropdown, allowing you to give another answer that you also think appropriate for that sentence. Do you want the second dropdown? ○ Yes ● No

Background: This paper details the creation and use of a massive (over 32,000 messages) artificially constructed 'Twitter' microblog stream for a regional emergency preparedness functional exercise.

Background: By combining microblog conversion, manual production, and a control set, we created a web based information stream providing valid, misleading, and irrelevant information to public information officers (PIOs) representing hospitals, fire departments, the local Red Cross, and city and county government officials.

Background: PIOs searched, monitored, and (through conventional channels) verified potentially acionable information that could then be redistributed through a personalized screen name.

Background: Our case study of a key PIO reveals several capabilities that social media can support, including event detection, the distribution of information between functions within the emergency response community, and the distribution of messages to the public.

Background: We suggest that training as well as information filtering tools are necessary to realize the potential of social media in both emergencies and exercises.

Table 1 Number of abstracts and sentences for each dataset by usage (i.e., training, validating and testing)

Dataset		Training	Validation	Testing
PubMed 20k	Nr. abstracts	15,000	2500	2500
	Nr. sentences	180,000	30,000	30,000
CS Abstracts	Nr. abstracts	500	77	77
	Nr. sentences	3287	824	619

PubMed 20k dataset, it is made publicly available with the data already partitioned as shown in Table 1. Figure 3 highlights the different distributions of the two datasets. Such result can be justified by the different data collection approaches and subsequent writing styles of both sciences, biomedical and computer science. Myers et al. [31] corroborate our claim by highlighting that the writing style is interconnected to the research method. This result supports the importance of our crowdsourcing approach in gathering a new dataset from another domain to validate our classifier, which is emphasized by the lack of publicly available classified datasets. Figure 3 also confirms that the different train, validation and test partitions have similar class distributions, which is a desired trait for classification tasks [30].

3.3 Neural networks models

In the last years, there have been remarkable developments in deep learning [11, 32]. Architectures such as convolutional neural network (CNN), LSTM and GRU have obtained competitive results in several competitions (e.g., computer vision, signal and natural language processing).

The CNN is a network mainly composed by convolutional layers. The purpose of the convolutional layers is to extract features that preserve relevant information from the inputs [33]. To obtain the features, a convolutional layer receives a matrix as input, to which a matrix with a set of weights, known as a filter, is applied using a sliding window approach and, at each of the sliding window steps, a convolution is calculated, resulting in a feature. The size of the filter is a relevant hyperparameter.

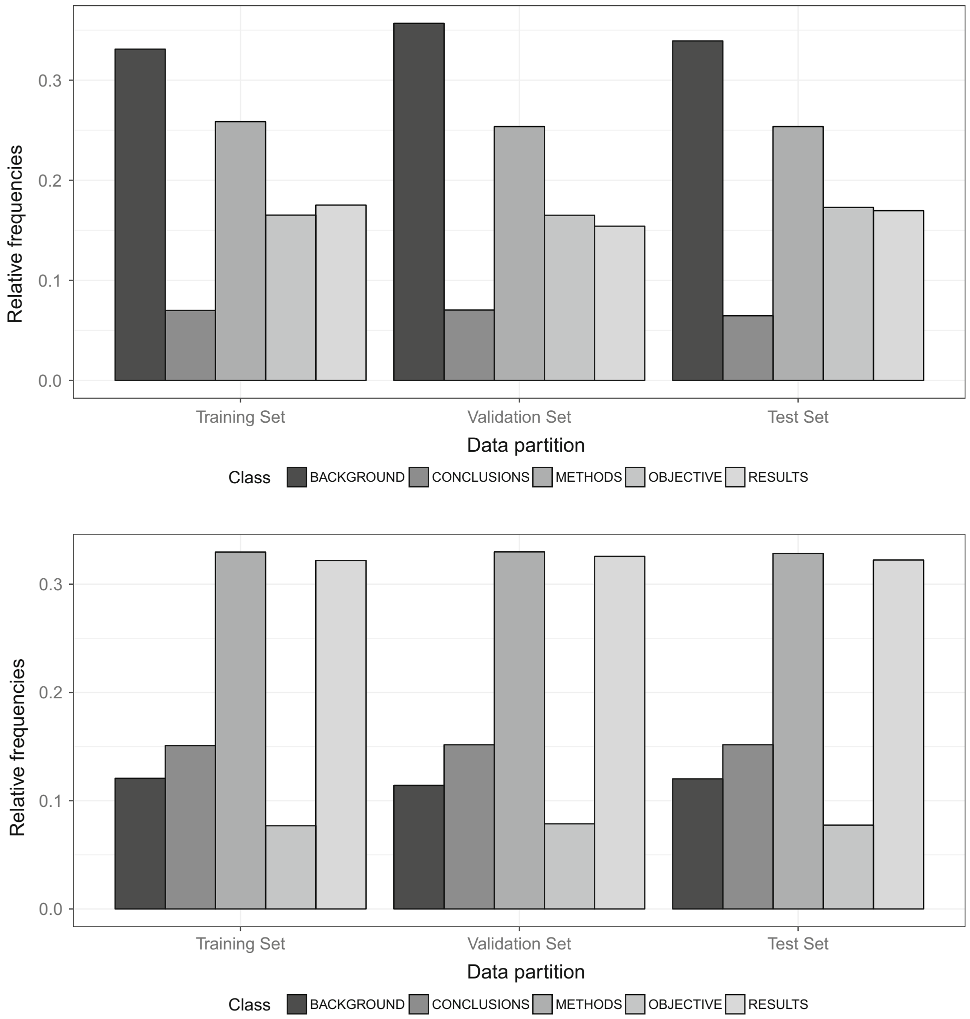


Fig. 3 Relative class frequencies for the CS Abstracts (top bar charts) and PubMed 20k (bottom bar chars)

Although CNNs have been widely used in computer vision, they can also be used in sentence classification [34]. The use of convolutional layers enables the extraction of features from a window of words, which is useful because word embeddings alone are not able to detect specific nuances, such as double negation, which is important for sentiment classification. The width of the filter, represented by h, determines the length of the n-grams. The number of filters is also a hyperparameter, making it possible to use multiple filters with varying lengths [34]. The filters are initialized with random weights, and, during network training, weights are learned for the specific task of the network, through backpropagation. Since each filter produces its own feature map, there is a need to reduce the dimensionality caused by using multiple filters. A sentence can be encoded as a single vector by applying a max pooling layer after the convolutional layer, which takes the maximum value for each position, from all the feature maps, keeping only the most important features.

Recurrent neural networks (RNN) are relevant for sequential data, such as the words that appear in a sentence. Consider the words $(x_1, \ldots, x_t)$ from a given sentence (sequence of words). The hidden state s_t of the word x_t depends on the hidden state s_{t-1}, which in turn is the hidden state of the word x_{t-1}, and, for this reason, the order in which words appear over the sequence also influences the various hidden states of the RNN.

The LSTM network is a particular RNN that uses an internal memory to keep information between distant time steps to model long-term dependencies of the sequence. It uses three gating mechanisms, input gate, forget gate and output gate, which control (at each hidden state), what new information should be updated into the memory, and what information should be erased from the memory. The GRU [35] was recently introduced, and it can be used as an alternative to the LSTM model. The GRU uses a reset and update gates, which are able to control how much information should be kept from previous time steps. Both GRU and LSTM are solutions that help mitigate the vanishing gradient problem of conventional RNNs.

A deep learning model was used by Dernoncourt et al. [5] for abstract sentence classification. The model uses character embeddings that are then concatenated with word embeddings and used as input for a bidirectional LSTM layer, which outputs a sentence vector based on those hybrid embeddings. The sentence vector is used to predict the probabilities of the labels for that sentence. The authors also use a sequence optimization layer, which has the objective of optimizing the classification of a sequence of sentences, exploiting existing dependencies between labels.

3.4 Proposed architecture

The proposed word embedding, convolutional and bidirectional GRU (Word-BiGRU) architecture is shown in Fig. 4. We assume that each abstract has i sentences $(S_1, \ldots, S_i)$ and each individual sentence has n words $(x_1^1, \ldots, x_n^i)$, where x_n^i is the nth word from the ith sentence. The various words from the sentences are mapped to their respective word embeddings, and those embedding are used to create a sentence matrix $E \in R^{m \times d}$, where d equals to the dimensionality of the embeddings. We use word embeddings pre-trained on English Wikipedia, provided by Glove (with $d = 200$) [36].

Then, a convolutional layer is used with a sliding window approach that extracts the most important features from the sentences. Let $E \in R^{m \times d}$ denote the sentence matrix, $w \in R^{h \times d}$ a filter, and $E[i : j]$ the sub-matrix from row i to j. The single feature o_i is obtained using:

$$o_i = w * E[i : i + h - 1]. \tag{1}$$

In this study, we use a filter with a size of $h = 5$. To add nonlinearity to the output, an activation function applied to every single feature. For the feature o_i, it is obtained by:

$$c_i = f(o_i + b); \tag{2}$$

where f is the activation function and b is the bias. We use ReLU as the activation function in our model because it tends to present a faster convergence [37].

Next, we take the various features maps obtained from the convolutional layer and feed them into a max pooling layer to encode the most important features extracted by the convolutional layer into a single vector representation that can be used by the next layers. Let $g_1, \ldots, g_i$ denote several vectors, each one encoding a particular sentence of the abstract. The vectors are then fed to bidirectional GRU layer, where the hidden states for each time step are calculated. We will use $\odot$ to denote the Hadamard Product, while using W and U to denote weight matrices of the GRU layer. Let h_{i-1} be the hidden state of the previous sentence from the same abstract, the candidate hidden state $\tilde{h}_i$ for the current sentence is given by:

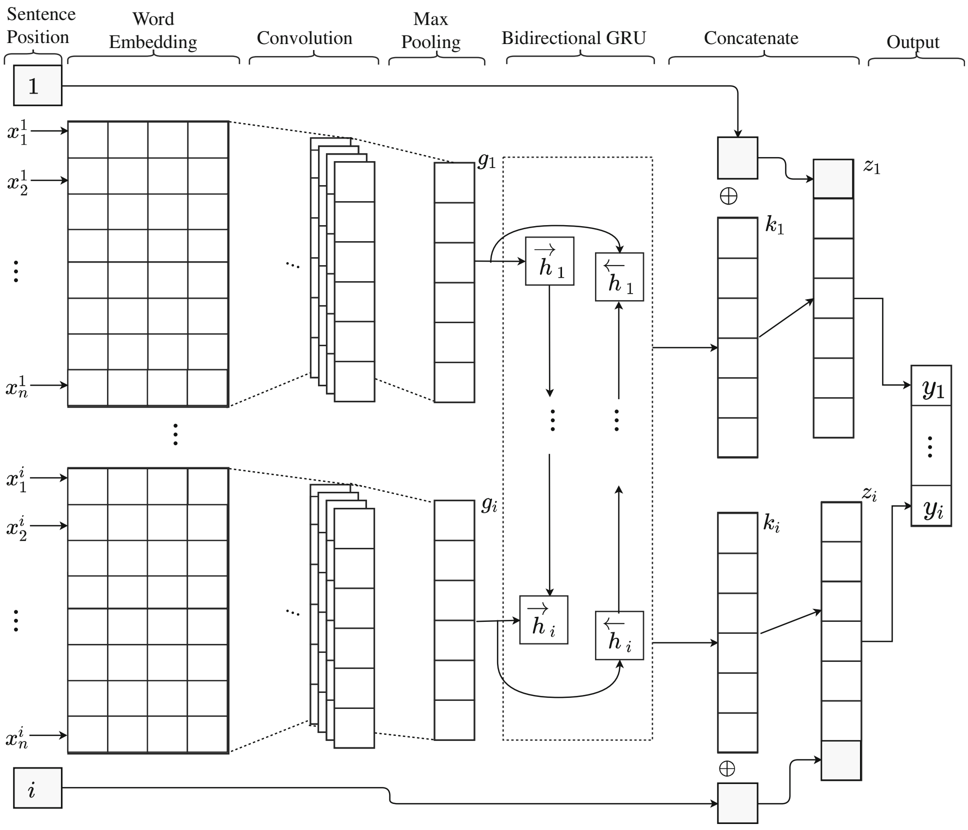


Fig. 4 Schematic of the proposed Word-BiGRU deep learning architecture

$$\tilde{h}_i = \tanh(W_h g_i + U_h(r_i \odot h_{i-1}) + b_h). \tag{3}$$

The reset gate $r_i \in [0,1]$ has the purpose of controlling how much information of the past hidden state, h_{t-1} will be kept. Let σ be the activation function, the reset gate r_i is calculated by:

$$r_i = \sigma(W_r g_i + U_r h_{i-1} + b_r). \tag{4}$$

To control how much new information will be stored in the hidden state, an update gate $z_i \in [0,1]$ is used, given by:

$$z_i = \sigma(W_z g_i + U_z h_{i-1} + b_z). \tag{5}$$

The hidden state h_i, which is the hidden state of the sentence i, is obtained by:

$$h_i = z_i \odot \tilde{h}_i + (1 - z_i) \odot h_{i-1}. \tag{6}$$

Since we use a bidirectional GRU layer, there is a forward pass and a backward pass. The hidden states resulting from the forward pass are:

$$(\overrightarrow{h_1}, \ldots, \overrightarrow{h_i}). \tag{7}$$

where h_i is the hidden state of the ith sentence of the abstract. Similarly, the hidden states resulting from the backward pass are:

$$(\overleftarrow{h_1}, \ldots, \overleftarrow{h_i}). \tag{8}$$

By using a bidirectional GRU, we want to capture contextual information about each sentence of the abstract, by taking into consideration the sentences that appear before and after it. For the ith sentence of the abstract, the individual vector k_i, which encodes the sentence with contextual information captured using the bidirectional GRU layer, is obtained by concatenating the forward and backward hidden states:

$$k_i = [\overrightarrow{h_i} \oplus \overleftarrow{h_i}]; \tag{9}$$

where $\oplus$ is the concatenation operator. Each encoded sentence k_i is then concatenated with an integer value indicating the position of that sentence in the abstract, resulting in z_i:

$$z_i = [k_i \oplus i]. \tag{10}$$

Finally, a softmax layer is used, such that the outputs can be interpreted as class probabilities.

3.5 Evaluation

In multiclass tasks, a classifier often outputs a class probability and the highest probability class is assigned as the predicted class label. Using these predicted labels, classification accuracy is often measured by building a confusion matrix, which maps predicted versus desired labels. From this matrix, several metrics can be computed, such as [38]: Precision, Recall, F1-score. For a class c, these metrics are obtained using:

$$\begin{aligned} \text{Precision}_c &= \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \\ \text{Recall}_c &= \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \\ \text{F1-score}_c &= 2 \times \frac{\text{Precision}_c \times \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \end{aligned} \tag{11}$$

where TP_c, FP_c, FN_c denote the number of true positives, false positives and false negatives for class c.

Another possibility to analyze multiclass probabilities is to consider one class probability p_c and a decision threshold $K \in [0,1]$. The class is considered positive if $p_k > K$. The receiver operating characteristic (ROC) curve shows the performance of the classifier across all K values for class c, plotting one minus the specificity (x-axis) versus the sensitivity (y-axis) [39]. The overall discriminatory performance is given by the area under the curve ($AUC = \int_0^1 ROCdK$). It is common to interpret the quality of the AUC values as: 0.5—equal to a random classifier; 0.6—reasonable; 0.7—good; 0.8—very good; 0.9—excellent; and 1—perfect.

To combine all five class confusion matrices results into a single measure, we adopt two aggregation methods: macro-averaging and weight-averaging. The macro-averaging computes first the metric (e.g., Precision using Eq. 11) for each class and then averages the overall result. The weight-averaging is computed in a similar way except that each class metric is weighted proportionally to its prevalence in the data. Dernoncourt et al. [5] used only the weight-averaging method. As for the ROC analysis, the five AUC class values are aggregated by using a macro-average aggregation method [40].

For comparison purposes, we adopt the same train, validation and test sets used by Dernoncourt et al. [5] for the PubMed 20k corpus. The respective set partition numbers are shown in Table 1. This table also includes the new CS abstract corpus training, validation and test division adopted numbers that can be used in future comparison works.

4 Results

4.1 Hyperparameter tuning

A predictive model needs to be tuned through an array of possible hyperparameters which may have a positive or negative impact in the model's predictive performance

[41]. However, finding the best combination of hyperparameters which optimizes a model's performance is a complex task, given the possible combinations to be tested and the required time to train a model. Also, some hyperparameters are encompassed within a numeric interval, which makes it impossible to test all values. In this section, we report the effort made in tuning the Word-BiGRU model using the PubMed 20k dataset, which is a larger dataset for which there are previous baseline results provided by Dernoncourt et al. [5].

The Word-BiGRU model shares some components with the CNN, such as the convolutional layer structure. Nevertheless, the Word-BiGRU model training takes significantly more time when compared to the CNN. Thus, the hyperparameters optimization for the convolutional layer was done using the CNN model to speed up the process. Therefore, the goal is to find the best values for the hyperparameters of the CNN model and then use those values in the convolutional layer of the Word-BiGRU model.

The filter size, number of filters and dropout are the main hyperparameters of the CNN model. The filter size is accountable for finding how many words within a sentence are encompassed by the filter in the sliding window process for the convolution operation. The number of filters represents how many filters are used in the convolutional layer, with each filter being capable of extracting different features. The dropout is used to reduce overfitting to training data [42]. Thus, the following configuration was considered as a baseline for the CNN model: filter size—32; number of filters—3; dropout—0.1. Also, to limit the space of possible solutions, the hyperparameter values were searched within the following intervals: filter size $\in \{3, 5, 8\}$; number of filters $\in \{32, 64, 128, 256\}$; dropout $\in \{0.1, 0.35, 0.5\}$.

Using such baseline, we explored the combinations of possible values. First, we changed the values of filter size, followed by the number of filters and, finally, by the dropout hyperparameter. Thus, this means that by exploring the number of filters, the best filter size value is already applied, since it was the first to be explored. According to Ng [30], hyperparameter selection should be done using one selected metric measured on the validation set. We follow such approach, in which we selected the weight-averaging method to aggregate the individual class metrics. The selected hyperparameter values are shown in Table 2. Thus, the best results for CNN within the defined intervals were obtained with 128 filters, each with size equals to 5, using a dropout of 0.35 after the convolutional layer.

For the specific GRU bidirectional layer of Word-BiGRU, we assumed the previously selected CNN hyperparameters and ranged the number of GRU units within {25, 50, 75, 100}, as shown in Table 3. Thus, the selected number of GRU units was 50. The final adopted configuration of the Word-BiGRU model, used for both PubMed 20k and CS Abstracts corpora, includes: number of filters—128; filter size—5; dropout—0.35; and number of GRU units—50.

Table 2 CNN model hyperparameters (in %, validation set results, best values in bold)

Hyperparameter	Valor	Precision	Sensitivity	F1-score
Number of filters	32	82.0	81.9	81.9
	64	82.6	82.3	82.4
	128	**83.3**	**83.4**	**83.3**
	256	83.2	83.2	83.2
Filter size	3	83.3	83.4	83.3
	5	**83.5**	**83.5**	**83.5**
	8	82.9	82.2	82.5
Dropout	0.1	83.5	83.5	83.5
	0.35	**83.6**	**83.6**	**83.6**
	0.5	81.5	80.6	81.0

Table 3 Word-BiGRU model hyperparameters (in %, validation set results, best results in bold)

Hyperparameter	Value	Precision	Sensitivity	F1-score
Number of units	25	89.4	89.3	89.3
	50	**91.5**	**91.5**	**91.5**
	75	89.8	89.7	89.7
	100	90.0	90.0	90.0

4.2 Biomedical abstracts

The performance in the PubMed 20k dataset can be compared with the results achieved by Dernoncourt et al. [5] (Char-BiLSTM). The latter uses word characters as inputs of a bidirectional LSTM layer, which provides vectorial representations of those words. Each vector is then used as input of another bidirectional LSTM layer which, in turn, provides a vectorial representation of a sentence. Additionally, Dernoncourt et al. [5] defined a transition matrix which contains the probabilities of transition between classes. The values of this matrix are computed through the model which, given a sentence being classified to a class (e.g., "Objective"), can then compute the probability of the next sentence belonging to another class (e.g., "Methods"). The results of our approach (Word-BiGRU) are also compared to the standard CNN model (described in Sect. 3.3) in Table 4. The best results for each metric are shown in bold. Our approach (Word-BiGRU) outperforms the alternatives for all computed metrics.

Table 4 Averaged test results for PubMed 20k (in %, best values in bold)

Metric	Averaged	Char-BiLSTM [5]	CNN	Word-BiGRU
Precision	Macro-averaged	86.4	80.7	**86.7**
	Weight-averaged	90.1	83.6	**90.9**
Recall	Macro-averaged	83.7	77.6	**86.7**
	Weight-averaged	89.9	83.5	**90.8**
F1-score	Macro-averaged	85.0	78.5	**86.7**
	Weight-averaged	90.0	83.5	**90.8**

The obtained results provide evidence that including contextual information improves performance for the PubMed 20k dataset. This is the main difference between the CNN and the Word-BiGRU, with the latter including contextual information through the GRU bidirectional layer. Such difference had a significant impact in all metrics.

The Word-BiGRU systematically outperforms the Char-BiLSTM model in all metrics, with improvements ranging from 0.3 (Precision macro-averaged) to 1.7 (F1-score macro-averaged) percentage points. Figure 5 shows the ROC curves, and respective AUC values (area), for each class. The average value computed through macro-average is 0.99. This is very close to 1, suggesting the model has an excellent discriminatory capability.

4.3 Computer science abstracts

Table 5 shows the classification performance test metrics for our Word-BiGRU approach in comparison with the CNN model for the CS Abstracts corpus. The World-BiGRU results are clearly better when compared with the CNN model, meaning that including contextual information through the bidirectional GRU layer had a significant impact on the model, confirming the results also obtained for the PubMed 20k dataset. When comparing the Word-BiGru results for both datasets, lower metric values were achieved for CS Abstracts (around 75%) when compared with PubMed 20k (around 91%). The classification performance differences can be due to several factors. For instance, PubMed 20k sentences were classified by the own authors, while the CS Abstracts were labeled using anonymous volunteers, via crowdsourcing. Furthermore, the PubMed 20k is around five times larger in size when compared to the CS Abstracts dataset and often deep learning models improve results with big data. This last issue is further explored in Sect. 4.4.

Table 5 Averaged test results for CS Abstracts (in %, best values in bold)

Metric	Method	CNN	Word-BiGRU
Precision	Macro-average	61.3	**75.4**
	Weight-averaged	66.3	**76.3**
Recall	Macro-averaged	57.0	**64.7**
	Weight-averaged	67.7	**73.0**
F1-score	Macro-averaged	59.1	**69.6**
	Weight-averaged	67.0	**74.6**

The ROC curves shown in Fig. 6 highlight a different picture of Word-BiGRU results when compared to the same model applied to the PubMeds 20k dataset (Fig. 5). The “Conclusions” class is the one with the lowest AUC

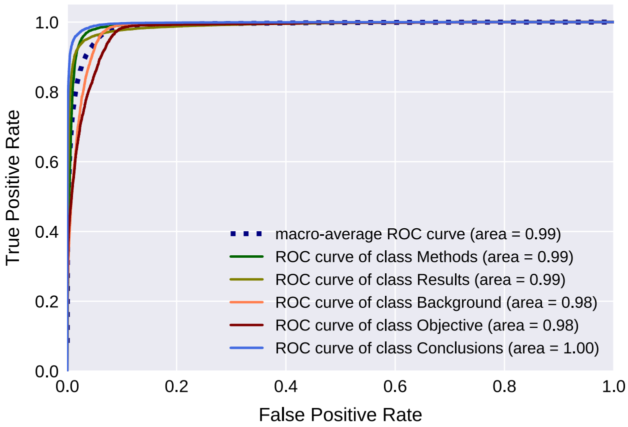


Fig. 5 ROC curves and AUCs for the Word-BiGRU model in the PubMed 20k dataset

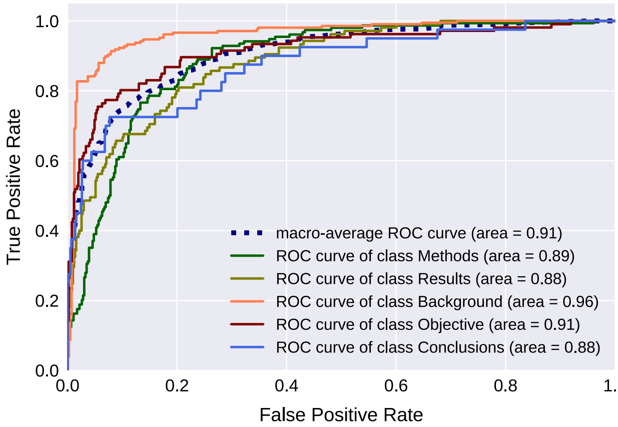


Fig. 6 ROC curves and AUCs for the Word-BiGRU model in the CS Abstracts dataset

(although still very good value of 88%), whereas in the PubMed 20k was the one with the highest AUC. Such result may derive from the different class distribution of both datasets (as shown in Fig. 3).

4.4 Data augmentation of CS abstracts

The CS Abstracts corpus has led to worse classification results when compared to the PubMed 20k data. Deep learning usually relies on large datasets for better training a model [43]. Thus, we argue that the CS Abstracts reduced size of 3287 sentences for training the model poses a limitation that we intend to overcome. A possibility that has recently been explored is data augmentation, which aims to artificially increase the size of the training dataset [44]. In domains such as computational visualization, simple transformations such as translation or scale changes can be directly applied to the training images to create different images that hold the same information (i.e., classified under the same category) [45].

However, in text mining, simple transformations are not a viable solution to data augmentation since textual data has a specific nature in which changing the form of a word may render a sentence to have a completely different meaning. A possible solution to overcome this issue is to replace words within sentences by synonyms. For example, a list of synonyms implemented in some NLP tools may be used, as demonstrated by Zhang et al. [46]. An alternative is to adopt translation to compute sentences with the same meaning, as shown by Pavel Ostyakov on GitHub.[2] The procedure starts by translating a sentence to other language and then back to the original language. This approach has the advantage of changing several word combinations at the same time and word order within a sentence, while keeping the same meaning, helping to build a more diverse training corpus. As such, we adopted this latter approach using the Textblob Python package. Also, we chose Spanish as the in-between translation language since the adopted NLP package was validated with success using the same language [47]. Table 6 shows three CS Abstracts sentences in both the original and the transformed formats. It shows that the sentences are slightly changed while retaining the same meaning.

This data augmentation process was only applied to the training set, which doubled in size from 3287 to 6574 sentences (i.e., for each sentence an equivalent one with the same meaning was added to the dataset). Table 7 shows the results of training the Word-BiGRU model with both the original and the data augmented datasets.

Using an augmented training set, when compared with the original dataset, leads to a decrease in Precision but an improvement in both Recall and F1-score measures. As for the ROC curves (Fig. 7), the obtained results are quite similar, with the augmented trained model presenting slight AUC improvements for the Methods and Results classes, with percentage point differences of: 1—“Methods” and 2—“Results.”

4.5 Sensitivity analysis

All the models evaluated are based on deep learning, which are considered black-box models. One way to open these black-box models is to use a sensitivity analysis [49], which monitors the output responses when varying the input features through their range of possible values. Specifically for textual contents, Ribeiro et al. [50] have developed the Lime package in Python, which changes model’s input by adding and removing words to assess the impact on the model’s output. We have adopted this package to obtain the words in sentences that influence the most each category. The aim is to understand the differences in the decision making process of our approach (Word-BiGRU) in comparison with CNN.

For demonstration purposes, in this section we compare the sensitivity analysis of CNN and Word-BiGRU models when trained with the augmented CS Abstracts dataset. Figure 8 shows the obtained output, when using the Lime package, for one selected abstract. Words shaded in red are the ones that contributed the most to classifying the sentence as “Background,” with particular emphasis for “media” and “development.” The words shaded in green are those that provided additional hints that the sentence may belong to other class than “Background.” The words shaded that belong to other sentences emerge because the Lime package affects the whole abstract.

Figure 9 shows the Lime results for the Word-BiGRU model on the same selected abstract. The model has a stronger confidence that the first sentence is related with “Background.” And some of the highlighted words for reaching to such conclusion are different (e.g., “latent attributes” instead of “development”). Another interesting feature of Word-BiGRU is that it uses several words from other sentences, as shown by looking at the shaded words in red in the sentences following the first. This is an evidence of the contextual information used for feeding the model.

The results of applying the Lime package to the CNN model for the second sentence of the abstract are shown in Fig. 10. Again, the sentence is accurately classified as “Background.” The words “however” and “current” were the ones influencing the most such classification.

[2] https://github.com/PavelOstyakov/toxic/blob/master/tools/extend_dataset.py.

Table 6 Comparison of original versus transformed sentences as a result of data augmentation

Original sentence	Transformed sentence
Early detection of such compromised accounts is very important in order to control the damage [48]	The early detection of such compromised accounts is very important to control the damage
In this work, we propose a novel general framework for discovering compromised accounts by utilizing statistical text analysis [48]	In this paper, we propose a new general framework for discovering compromised accounts through the use of statistical text analysis
These are the anomalies caused by a user because of his/her variable behavior toward different sources [48]	These are the anomalies caused by a user due to their variable behavior toward different sources

Table 7 Comparative results between original and data augmented datasets for the Word-BiGRU model (best values in bold)

Metric	Method	Original	Data augmented
Precision	Macro-averaged	**75.4**	69.7
	Weight-averaged	**76.3**	75.4
Recall	Macro-averaged	64.7	**70.0**
	Weight-averaged	73.0	**74.9**
F1-score	Macro-averaged	69.6	**69.8**
	Weight-averaged	74.6	**75.1**

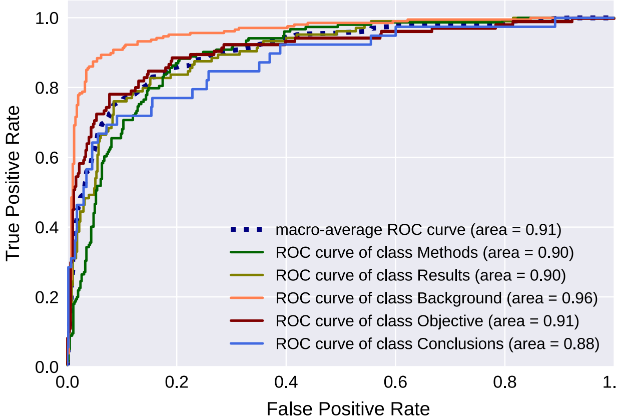


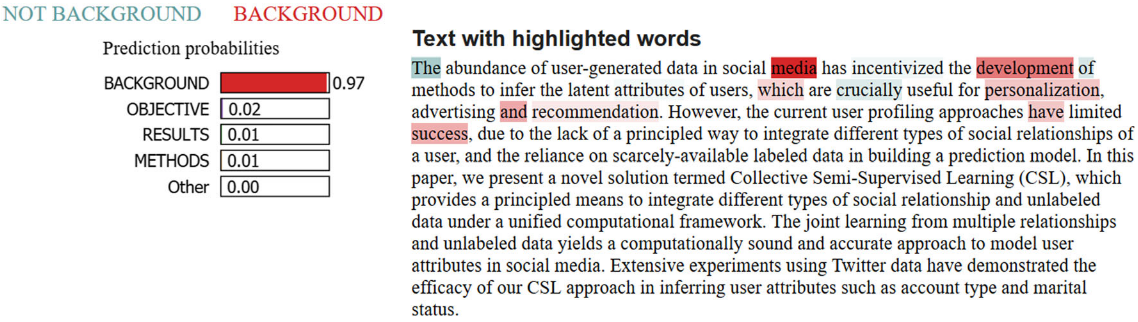
Fig. 7 ROC curves and AUCs for the Word-BiGRU model in the original and data augmented CS Abstracts datasets

For the Word-BiGRU model, the probability of the second sentence being classified as "Background" is higher, again proving the value of the contextual information included in the model (Fig. 11). This can be shown in the highlighted words in sentences other than the second.

5 Conclusions

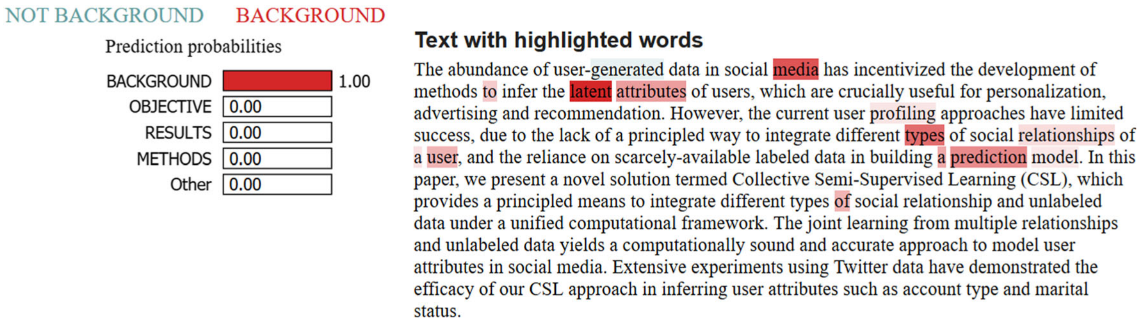
This paper presents a novel deep learning architecture for the classification of scientific abstract sentences (background, objectives, methods, results, conclusions), which is valuable to assist in scientific database querying, performing literature reviews and to support the writing of new abstracts. The proposed Word-BiGRU architecture assumes word embeddings, a convolutional layer and a bidirectional gated recurrent unit (GRU). Using a large sentence corpus, related with 20k abstracts from the biomedical domain, we have obtained high quality classification performances, with weight-averaged Precision, Recall and F1-score values around 91%. These results compare favorably against a state-of-the-art bidirectional long short-term memory (LSTM) model.

We also collected a new dataset of Computer Science scientific abstracts using a crowdsourcing approach to assess the performance of our architecture in a new domain. The results in this 4k classified sentences dataset were below the ones achieved for the 20k biomedical dataset when using the same Word-BiGRU architecture. However, a very good classification level was achieved by



The abundance of user-generated data in social media has incentivized the development of methods to infer the latent attributes of users, which are crucially useful for personalization, advertising and recommendation. However, the current user profiling approaches have limited success, due to the lack of a principled way to integrate different types of social relationships of a user, and the reliance on scarcely-available labeled data in building a prediction model. In this paper, we present a novel solution termed Collective Semi-Supervised Learning (CSL), which provides a principled means to integrate different types of social relationship and unlabeled data under a unified computational framework. The joint learning from multiple relationships and unlabeled data yields a computationally sound and accurate approach to model user attributes in social media. Extensive experiments using Twitter data have demonstrated the efficacy of our CSL approach in inferring user attributes such as account type and marital status.

Fig. 8 Most relevant words in classifying the first sentence for CNN model


NOT BACKGROUND BACKGROUND

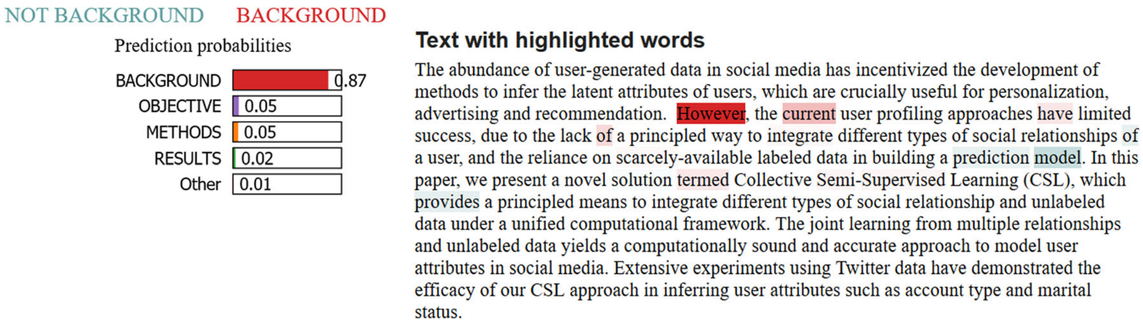
Prediction probabilities

BACKGROUND	1.00
OBJECTIVE	0.00
RESULTS	0.00
METHODS	0.00
Other	0.00

Text with highlighted words

The abundance of user-generated data in social media has incentivized the development of methods to infer the latent attributes of users, which are crucially useful for personalization, advertising and recommendation. However, the current user profiling approaches have limited success, due to the lack of a principled way to integrate different types of social relationships of a user, and the reliance on scarcely-available labeled data in building a prediction model. In this paper, we present a novel solution termed Collective Semi-Supervised Learning (CSL), which provides a principled means to integrate different types of social relationship and unlabeled data under a unified computational framework. The joint learning from multiple relationships and unlabeled data yields a computationally sound and accurate approach to model user attributes in social media. Extensive experiments using Twitter data have demonstrated the efficacy of our CSL approach in inferring user attributes such as account type and marital status.

Fig. 9 Most relevant words in classifying the first sentence for Word-BiGRU model


NOT BACKGROUND BACKGROUND

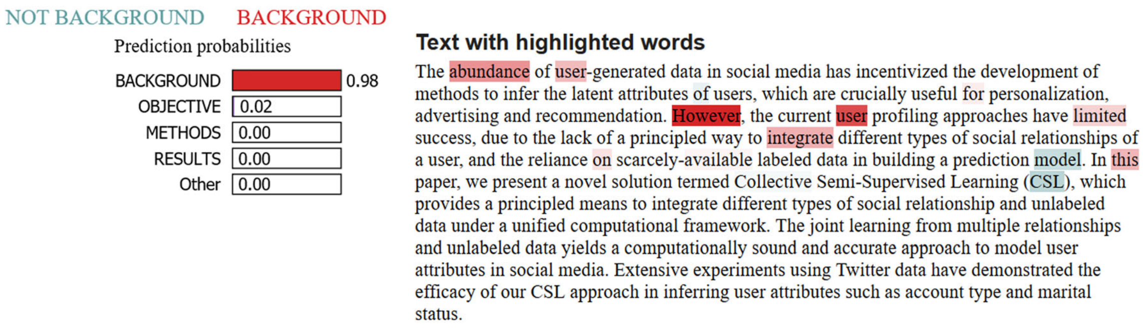
Prediction probabilities

BACKGROUND	0.87
OBJECTIVE	0.05
METHODS	0.05
RESULTS	0.02
Other	0.01

Text with highlighted words

The abundance of user-generated data in social media has incentivized the development of methods to infer the latent attributes of users, which are crucially useful for personalization, advertising and recommendation. However, the current user profiling approaches have limited success, due to the lack of a principled way to integrate different types of social relationships of a user, and the reliance on scarcely-available labeled data in building a prediction model. In this paper, we present a novel solution termed Collective Semi-Supervised Learning (CSL), which provides a principled means to integrate different types of social relationship and unlabeled data under a unified computational framework. The joint learning from multiple relationships and unlabeled data yields a computationally sound and accurate approach to model user attributes in social media. Extensive experiments using Twitter data have demonstrated the efficacy of our CSL approach in inferring user attributes such as account type and marital status.

Fig. 10 Most relevant words in classifying the second sentence for CNN model


NOT BACKGROUND BACKGROUND

Prediction probabilities

BACKGROUND	0.98
OBJECTIVE	0.02
METHODS	0.00
RESULTS	0.00
Other	0.00

Text with highlighted words

The abundance of user-generated data in social media has incentivized the development of methods to infer the latent attributes of users, which are crucially useful for personalization, advertising and recommendation. However, the current user profiling approaches have limited success, due to the lack of a principled way to integrate different types of social relationships of a user, and the reliance on scarcely-available labeled data in building a prediction model. In this paper, we present a novel solution termed Collective Semi-Supervised Learning (CSL), which provides a principled means to integrate different types of social relationship and unlabeled data under a unified computational framework. The joint learning from multiple relationships and unlabeled data yields a computationally sound and accurate approach to model user attributes in social media. Extensive experiments using Twitter data have demonstrated the efficacy of our CSL approach in inferring user attributes such as account type and marital status.

Fig. 11 Most relevant words in classifying the second sentence for Word-BiGRU model

Word-BiGRU, which compared favorably with a convolutional neural network (CNN) model, obtaining weight-averaged Precision, Recall and F1-score values around 75%. Furthermore, we confirmed the decision process of Word-BiGRU and CNN through a sensitivity analysis procedure, and we found evidences of the former using the contextual information included through the GRU. To address the reduced size of the collected Computer Science abstracts dataset, we adopted a data augmentation approach based on sentence translation to Spanish and back to English, which resulted in a training corpus with twice the original size. The classification metrics on test set data have shown a slight improvement of the augmented trained Word-BiGru model in terms of sensitivity and F1-scores.

In future research, we intend to check if we can improve the Computer Science corpus results by further increasing the Computer Science corpus size, either by collecting more human labels or by combining our data augmentation translation technique with other synthetic data creation methods (e.g., usage of synonyms). We also wish to enlarge the experimentation of the proposed Word-BiGRU deep learning architecture to other sequential tasks.

Acknowledgements This work was supported by Fundação para a Ciência e Tecnologia (FCT) within the Project Scope: UID/CEC/00319/2019.

References

1. Michalska-Smith MJ, Allesina S (2017) And, not or: quality, quantity in scientific publishing. PLoS ONE 12(6):e0178074

2. Khabsa M, Giles CL (2014) The number of scholarly documents on the public web. PLoS ONE 9(5):e93949
3. Atanassova I, Bertin M, Larivière V (2016) On the composition of scientific abstracts. J Doc 72(4):636–647
4. Zubiaga Arkaitz, Kochkina E, Liakata M, Procter R, Lukasik M, Bontcheva K, Cohn T, Augenstein I (2018) Discourse-aware rumour stance classification in social media using sequential classifiers. Inf Process Manag 54(2):273–290. https://doi.org/10.1016/j.ipm.2017.11.009
5. Dernoncourt F, Lee JY, Szolovits P (2017) Neural networks for joint sentence classification in medical paper abstracts. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics: volume 2, short papers, vol 2, pp 694–700
6. Cornuel E (2005) A vision for business schools, vol 24. Emerald Group Publishing, Bingley
7. Kitchenham B, Brereton P (2013) A systematic review of systematic review process research in software engineering. Inf Softw Technol 55(12):2049–2075
8. Moro S, Cortez P, Rita P (2015) Business intelligence in banking: a literature analysis from 2002 to 2013 using text mining and latent Dirichlet allocation. Expert Syst Appl 42(3):1314–1324
9. Liu Y, Wu F, Liu M, Liu B (2013) Abstract sentence classification for scientific papers based on transductive svm. Comput Inf Sci 6(4):125
10. Dernoncourt F, Lee JY (2017) Pubmed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In: Proceedings of the eighth international joint conference on natural language processing (volume 2: short papers), vol 2, pp 308–313
11. Goodfellow I, Bengio Y, Courville A, Bengio Y (2016) Deep learning, vol 1. MIT Press, Cambridge
12. Boudin F, Nie JY, Bartlett JC, Grad R, Pluye P, Dawes M (2010) Combining classifiers for robust pico element detection. BMC Med Inform Decis Mak 10(1):29
13. Dellermann D, Ebel P, Söllner M, Leimeister JM (2019) Hybrid intelligence. Bus Inf Syst Eng. https://doi.org/10.1007/s12599-019-00595-2
14. Tsapatsoulis N, Djouvas C (2019) Opinion mining from social media short texts: does collective intelligence beat deep learning? Front Robot AI. https://doi.org/10.3389/frobt.2018.00138
15. Brabham DC (2008) Crowdsourcing as a model for problem solving: an introduction and cases. Convergence 14(1):75–90
16. Welinder P, Perona P (2010) Online crowdsourcing: rating annotators and obtaining cost-effective labels. In: 2010 IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW). IEEE, pp 25–32
17. Morschheuser B, Hamari J, Koivisto J (2016) Gamification in crowdsourcing: a review. In: 2016 49th Hawaii international conference on system sciences (HICSS). IEEE, pp 4375–4384
18. Hossain M (2012) Users' motivation to participate in online crowdsourcing platforms. In: 2012 international conference on innovation management and technology research (ICIMTR). IEEE, pp 310–315
19. Massung E, Coyle D, Cater KF, Jay M, Preist C (2013) Using crowdsourcing to support pro-environmental community activism. In: Proceedings of the SIGCHI conference on human factors in computing systems, ACM, pp 371–380
20. Zheng H, Li D, Hou W (2011) Task design, motivation, and participation in crowdsourcing contests. Int J Electron Commer 15(4):57–88
21. Moro S, Ramos P, Esmerado J, Jalali SMJ (2019) Can we trace back hotel online reviews characteristics using gamification features? Int J Inf Manag 44:88–95
22. Canito J, Ramos P, Moro S, Rita P (2018) Unfolding the relations between companies and technologies under the big data umbrella. Comput Ind 99:1–8
23. Di Bitetti MS, Ferreras JA (2017) Publish (in English) or perish: the effect on citation rate of using languages other than English in scientific publications. Ambio 46(1):121–127
24. Feinerer I, Buchta C, Geiger W, Rauch J, Mair P, Hornik K (2013) The textcat package for n-gram based text categorization in R. J Stat Softw 52(6):1–17
25. Panettieri J (2017) Cloud market share 2017: Amazon aws, microsoft azure, ibm, google. ChannelE2E
26. Suciu G, Scheianu A, Vochin M (2017) Disaster early warning using time-critical iot on elastic cloud workbench. In: 2017 IEEE International Black Sea conference on communications and networking (BlackSeaCom). IEEE, pp 1–5
27. Stewart O, Lubensky D, Huerta JM (2010) Crowdsourcing participation inequality: a scout model for the enterprise domain. In: Proceedings of the ACM SIGKDD workshop on human computation. ACM, pp 30–33
28. Moro S, Laureano R, Cortez P (2011) Using data mining for bank direct marketing: an application of the crisp-dm methodology. In: Proceedings of European simulation and modelling conference-ESM'2011, EUROSIS-ETI, pp 117–121
29. Yuan X, Liao X, Li S, Shi Q, Wu J, Li K (2019) Extracting PICO elements from RCT abstracts using 1-2gram analysis and multitask classification. CoRR abs/1901.08351, arxiv:1901.08351
30. Ng A (2017) Machine learning yearning. Stanford Press
31. Myers MD et al (1997) Qualitative research in information systems. Manag Inf Syst Q 21(2):241–242
32. Zhang Q, Yang LT, Chen Z, Li P (2018) A survey on deep learning for big data. Inf Fusion 42:146–157
33. LeCun Y, Kavukcuoglu K, Farabet C (2010) Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE international symposium on circuits and systems, pp 253–256
34. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1746–1751
35. Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp 1724–1734
36. Pennington J, Socher R, Manning C (2014) GloVe: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
37. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp 315–323
38. Witten I, Frank E, Hall M, Pal C (2017) Data mining: practical machine learning tools and techniques, 4th edn. Morgan Kaufmann, San Franscico
39. Moro S, Cortez P, Rita P (2017) A framework for increasing the value of predictive data-driven models by enriching problem domain characterization with novel features. Neural Comput Appl 28(6):1515–1523
40. Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27:861–874
41. Stoean R (2018) Analysis on the potential of an EA-surrogate modelling tandem for deep learning parametrization: an example for cancer classification from medical images. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3709-5
42. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
43. Rebai I, BenAyed Y, Mahdi W (2016) Deep multilayer multiple kernel learning. Neural Comput Appl 27(8):2305–2314

44. Bawa VS, Kumar V (2018) Emotional sentiment analysis for a group of people based on transfer learning with a multi-modal system. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3867-5
45. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
46. Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. In: Advances in neural information processing systems. MIT Press, Cambridge, pp 649–657
47. Torres JP, de Piñerez Reyes RG, Bucheli VA (2018) Support vector machines for semantic relation extraction in Spanish language. In: Colombian conference on computing. Springer, pp 326–337
48. Seyler D, Li L, Zhai C (2018) Identifying compromised accounts on social media using statistical text analysis. arXiv preprint arXiv:180407247
49. Cortez P, Embrechts MJ (2013) Using sensitivity analysis and visualization techniques to open black box data mining models. Inf Sci 225:1–17
50. Ribeiro MT, Singh S, Guestrin C (2016) “why should I trust you?”: explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA, August 13–17, 2016, pp 1135–1144