frontiers in Research Metrics and Analytics

ORIGINAL RESEARCH
published: 10 February 2021
doi: 10.3389/frma.2020.600382

Large Scale Subject Category Classification of Scholarly Papers With Deep Attentive Neural Networks

Bharath Kandimalla[1], Shaurya Rohatgi[2], Jian Wu[3]* and C. Lee Giles[1,2]*

[1]Computer Science and Engineering, Pennsylvania State University, University Park, PA, United States, [2]Information Sciences and Technology, Pennsylvania State University, University Park, PA, United States, [3]Computer Science, Old Dominion University, Norfolk, VA, United States

Subject categories of scholarly papers generally refer to the knowledge domain(s) to which the papers belong, examples being computer science or physics. Subject category classification is a prerequisite for bibliometric studies, organizing scientific publications for domain knowledge extraction, and facilitating faceted searches for digital library search engines. Unfortunately, many academic papers do not have such information as part of their metadata. Most existing methods for solving this task focus on unsupervised learning that often relies on citation networks. However, a complete list of papers citing the current paper may not be readily available. In particular, new papers that have few or no citations cannot be classified using such methods. Here, we propose a deep attentive neural network (DANN) that classifies scholarly papers using only their abstracts. The network is trained using nine million abstracts from Web of Science (WoS). We also use the WoS schema that covers 104 subject categories. The proposed network consists of two bi-directional recurrent neural networks followed by an attention layer. We compare our model against baselines by varying the architecture and text representation. Our best model achieves micro-F_1 measure of 0.76 with F_1 of individual subject categories ranging from 0.50 to 0.95. The results showed the importance of retraining word embedding models to maximize the vocabulary overlap and the effectiveness of the attention mechanism. The combination of word vectors with TFIDF outperforms character and sentence level embedding models. We discuss imbalanced samples and overlapping categories and suggest possible strategies for mitigation. We also determine the subject category distribution in CiteSeerX by classifying a random sample of one million academic papers.

Keywords: text classification, text mining, scientific papers, digital library, neural networks, citeseerx, subject category classification

OPEN ACCESS

Edited by:
Philipp Mayr,
GESIS Leibniz Institute for the Social Sciences, Germany

Reviewed by:
Nansu Zong,
Mayo Clinic, United States
Frederique Bordignon,
École des ponts ParisTech, France

****Correspondence:***
Bharath Kandimalla
bharathkumarkandimalla@gmail.com
Jian Wu
jwu@cs.odu.edu

Specialty section:
This article was submitted to Text-mining and Literature-based Discovery, a section of the journal Frontiers in Research Metrics and Analytics

Received: *29 August 2020*
Accepted: *24 December 2020*
Published: *10 February 2021*

Citation:
Kandimalla B, Rohatgi S, Wu J and Giles CL (2021) Large Scale Subject Category Classification of Scholarly Papers With Deep Attentive Neural Networks. Front. Res. Metr. Anal. 5:600382. doi: 10.3389/frma.2020.600382

1 INTRODUCTION

A recent estimate of the total number of English research articles available online was at least 114 million (Khabsa and Giles, 2014). Studies indicate the number of academic papers doubles every 10–15 years (Larsen and von Ins, 2010). The continued growth of scholarly papers increases the challenges to accurately find relevant research papers, especially when papers in different subject categories (SCs) are mixed in a search engine's collection. Searches based on only keywords may no longer be the most efficient method

(Matsuda and Fukushima, 1999) to use. This often happens when the same query terms appear in multiple research areas. For example, querying "neuron" in Google Scholar returns documents in both computer science and neuroscience. Search results can also belong to diverse domains when the query terms contain acronyms. For example, querying "NLP" returns documents in linguistics (meaning "neuro-linguistic programming") and computer science (meaning "natural language processing"). If the SCs of documents are available, the users can narrow search results by specifying an SC, which effectively increases the precision of the query results, assuming SCs are accurately assigned to documents. Also, delineation of scientific domains is a preliminary tasks of many bibliometric studies at the meso-level. Accurate categorization of research articles is a prerequisite for discovering various dimensions of scientific activity in epistemology (Collins, 1992) and sociology (Barnes et al., 1996), as well as the invisible colleges, which are implicit academic networks (Zitt et al., 2019). To build a web-scale knowledge system, it is necessary to organize scientific publications into a hierarchical concept structure, which further requires categorization of research articles by SCs (Shen et al., 2018).

As such, we believe it is useful to build a classification system that assigns SCs to scholarly papers. Such a system could significantly impact scientific search and facilitate bibliometric evaluation. It can also help with Science of Science research (Fortunato et al., 2018), an area of research that uses scholarly big data to study the choice of scientific problems, scientist career trajectories, research trends, research funding, and other research aspects. Also, many have noted that it is difficult to extract SCs using traditional topic models such as Latent Dirichlet Allocation (LDA), since it only extracts words and phrases present in documents (Gerlach et al., 2018). An example is that a paper in computer science is rarely given its SC in the keyword list.

In this work, we pose the SC problem as one of multiclass classifications in which one SC is assigned to each paper. In a preliminary study, we investigated feature-based machine learning methods to classify research papers into six SCs (Wu et al., 2018). Here, we extend that study and propose a system that classifies scholarly papers into 104 SCs using only abstracts. The core component is a neural network classifier trained on millions of labeled documents that are part of the WoS database. In comparison with our preliminary work, our data is more heterogeneous (more than 100 SCs as opposed to six), imbalanced, and complicated (data labels may overlap). We compare our system against several baselines applying various text representations, machine learning models, and/or neural network architectures.

SC classification is usually based on a universal schema for a specific domain or for all domains. Many schemas for scientific classification systems are publisher domain specific. For example, ACM has its own hierarchical classification system[1], NLM has medical subject headings[2], and MSC has a subject classification for mathematics[3]. The most comprehensive and systematic classification schemas seem to be from WoS[4] and the Library of Congress (LOC)[5]. The latter was created in 1897 and was driven by practical needs of the LOC rather than any epistemological considerations and is most likely out of date.

To the best of our knowledge, our work is the first example of using a neural network to classify scholarly papers into a comprehensive set of SCs. Other work focused on unsupervised methods and most were developed for specific category domains. In contrast, our classifier was trained on a large number of high quality abstracts from WoS and can be applied directly to abstracts without any citation information. We also develop a novel representation of scholarly paper abstracts using ranked tokens and their word embedding representations. This significantly reduces the scale of the classic Bag of Word (BoW) model. We also retrained FastText and GloVe word embedding models using WoS abstracts. The subject category classification was then applied to the CiteSeerX collection of documents. However, it could be applied to any similar collection.

2 RELATED WORK

Text classification is a fundamental task in natural language processing. Many complicated tasks use it or include it as a necessary step, such as part-of-speech tagging, e.g., Ratnaparkhi (1996), sentiment analysis, e.g., Vo and Zhang (2015), and named entity recognition, e.g., Nadeau and Sekine (2007). Classification can be performed at many levels: word, phrase, sentence, snippet (e.g., tweets, reviews), articles (e.g., news articles), and others. The number of classes usually ranges from a few to nearly 100. Methodologically, a classification model can be supervised, semi-supervised, and unsupervised. An exhaustive survey is beyond the scope of this paper. Here we briefly review short text classification and highlight work that classifies scientific articles.

Bag of words (BoWs) is one of the most commonly used representations for text classification, an example being keyphrase extraction (Caragea et al., 2016; He et al., 2018). BoW represents text as a set of unordered word-level tokens, without considering syntactical and sequential information. For example, Nam et al. (2016) combined BoW with linguistic, grammatical, and structural features to classify sentences in biomedical paper abstracts. In Li et al. (2010), the authors treated the text classification as a sequence tagging problem and proposed a Hidden Markov Model used for the task of classifying sentences into mutually exclusive categories, namely, background, objective, method, result, and conclusions. The task described in García et al. (2012) classifies abstracts in biomedical databases into 23 categories (OHSUMED dataset) or 26 categories (UVigoMED dataset). The author proposed a bag-of-concept representation based on Wikipedia and classify abstracts using the SVM model.

[1]https://www.acm.org/about-acm/class
[2]https://www.ncbi.nlm.nih.gov/mesh
[3]http://msc2010.org/mediawiki/index.php?title=Main_Page
[4]https://images.webofknowledge.com/images/help/WOS/hp_subject_category_terms_tasca.html
[5]https://www.loc.gov/aba/cataloging/classification/

Recently, word embeddings (WE) have been used to build distributed dense vector representations for text. Embedded vectors can be used to measure semantic similarity between words (Mikolov et al., 2013b). WE has shown improvements in semantic parsing and similarity analysis, e.g., Prasad et al. (2018). Other types of embeddings were later developed for character level embedding (Zhang et al., 2015), phrase embedding (Passos et al., 2014), and sentence embedding (Cer et al., 2018). Several WE models have been trained and distributed; examples are word2vec (Mikolov et al., 2013b), GloVe (Pennington et al., 2014), FastText (Grave et al., 2017), Universal Sentence Encoder (Cer et al., 2018), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2019). Empirically, Long Short Term Memory [LSTM; Hochreiter and Schmidhuber (1997)], Gated Recurrent Units [GRU; Cho et al. (2014)], and convolutional neural networks [CNN; LeCun et al. (1989)] have achieved improved performance compared to other supervised machine learning models based on shallow features (Ren et al., 2016).

Classifying SCs of scientific documents is usually based on metadata, since full text is not available for most papers and processing a large amount of full text is computationally expensive. Most existing methods for SC classification are unsupervised. For example, the Smart Local Moving Algorithm identified topics in PubMed based on text similarity (Boyack and Klavans, 2018) and citation information (van Eck and Waltman, 2017). K-means was used to cluster articles based on semantic similarity (Wang and Koopman, 2017). The memetic algorithm, a type of evolutionary computing (Moscato and Cotta, 2003), was used to classify astrophysical papers into subdomains using their citation networks. A hybrid clustering method was proposed based on a combination of bibliographic coupling and textual similarities using the Louvain algorithm-a greedy method that extracted communities from large networks (Glänzel and Thijs, 2017). Another study constructed a publication-based classification system of science using the WoS dataset (Waltman and van Eck, 2012). The clustering algorithm, described as a modularity-based clustering, is conceptually similar to k-nearest neighbor (kNN). It starts with a small set of seed labeled publications and grows by incrementally absorbing similar articles using co-citation and bibliographic coupling. Many methods mentioned above rely on citation relationships. Although such information can be manually obtained from large search engines such as Google Scholar, it is non-trivial to scale this for millions of papers.

Our model classifies papers based only on abstracts, which are often available. Our end-to-end system is trained on a large number of labeled data with no references to external knowledge bases. When compared with citation-based clustering methods, we believe it to be more scalable and portable.

3 TEXT REPRESENTATIONS

For this work, we represent each abstract using a BoW model weighted by TF-IDF. However, instead of building a sparse vector for all tokens in the vocabulary, we choose word tokens with the highest TF-IDF values and encode them using WE models. We explore both pre-trained and re-trained WE models. We also explore their effect on classification performance based on token order. As evaluation baselines, we compare our best model with off-the-shelf text embedding models, such as the Unified Sentence Encoder [USE; Cer et al. (2018)]. We show that our model which uses the traditional and relatively simple BoW representation is computationally less expensive and can be used to classify scholarly papers at scale, such as those in the CiteSeerX repository (Giles et al., 1998; Wu et al., 2014).

3.1 Representing Abstracts

First, an abstract is tokenized with white spaces, punctuation, and stop words were removed. Then a list $\mathbf{A}$ of word types (unique words) w_i is generated after lemmatization which uses the WordNet database (Fellbaum, 2005) for the lemmas.

$$\mathbf{A} = [w_1, w_2, w_3 \ldots w_n]. \tag{1}$$

Next the list $\mathbf{A}_f$ is sorted in descending order by TF-IDF giving $\mathbf{A}_{\text{sorted}}$. TF is the term frequency in an abstract and IDF is the inverse document frequency calculated using the number of abstracts containing a token in the entire WoS abstract corpus.

$$\mathbf{A}_{\text{sorted}} = \left[w_1', w_2', w_3' \ldots w_n'\right]. \tag{2}$$

Because abstracts may have different numbers of words, we chose the top d elements from $\mathbf{A}_{\text{sorted}}$ to represent the abstract. We then re-organize the elements according to their original order in the abstract forming a sequential input. If the number of words is less than d, we pad the feature list with zeros. The final list is a vector built by concatenating all word level vectors $\vec{v}'_k, k \in \{1, \cdots, d\}$ into a D_{WE} dimension vector. The final semantic feature vector $\mathbf{A}_f$ is:

$$\mathbf{A}_f = \left[\vec{v}'_1, \vec{v}'_2, \vec{v}'_3 \ldots \vec{v}'_d\right] \tag{3}$$

3.2 Word Embedding

To investigate how different word embeddings affect classification results, we apply several widely used models. An exhaustive experiment for all possible models is beyond the scope of this paper. We use some of the more popular ones as now discussed.

GloVe captures semantic correlations between words using global word-word co-occurrence, as opposed to local information used in word2vec (Mikolov et al., 2013a). It learns a word-word co-occurrence matrix and predicts co-occurrence ratios of given words in context (Pennington et al., 2014). Glove is a context-independent model and outperformed other word embedding models such as word2vec in tasks such as word analogy, word similarity, and named entity recognition tasks.

FastText is another context-independent model which uses sub-word (e.g., character n-grams) information to represent words as vectors (Bojanowski et al., 2017). It uses log-bilinear models that ignore the morphological forms by assigning distinct vectors for each word. If we consider a word w whose n-grams

are denoted by g_w, then the vector $\mathbf{z}_g$ is assigned to each n-gram in g_w. Each word is represented by the sum of the vector representations of its character n-grams. This representation is incorporated into a Skip Gram model (Goldberg and Levy, 2014) which improves vector representation for morphologically rich languages.

SciBERT is a variant of BERT, a context-aware WE model that has improved the performance of many NLP tasks such as question answering and inference (Devlin et al., 2019). The bidirectionally trained model seems to learn a deeper sense of language than single directional transformers. The transformer uses an attention mechanism that learns contextual relationships between words. SciBERT uses the same training method as BERT but is trained on research papers from Semantic Scholar. Since the abstracts from WoS articles mostly contain scientific information, we use SciBERT (Beltagy et al., 2019) instead of BERT. Since it is computationally expensive to train BERT (4 days on 4–16 Cloud TPUs as reported by Google), we use the pre-trained SciBERT.

3.3 Retrained WE Models

Though pretrained WE models represent richer semantic information compared with traditional one-hot vector methods, when applied to text in scientific articles the classifier does not perform well. This is probably because the text corpus used to train these models are mostly from Wikipedia and Newswire. The majority of words and phrases included in the vocabulary extracted from these articles provides general descriptions of knowledge, which are significantly different from those used in scholarly articles which describe specific domain knowledge. Statistically, the overlap between the vocabulary of pretrained GloVe (six billion tokens) and WoS is only 37% (Wu et al., 2018). Nearly all of the WE models can be retrained. Thus, we retrained GloVe and FastText using 6.38 million abstracts in WoS (by imposing a limit of 150k on each SC, see below for more details). There are 1.13 billion word tokens in total. GloVe generated 1 million unique vectors, and FastText generated 1.2 million unique vectors.

3.4 Universal Sentence Encoder

For baselines, we compared with Google's Universal Sentence Encoder (USE) and the character-level convolutional network (CCNN). USE uses transfer learning to encode sentences into vectors. The architecture consists of a transformer-based sentence encoding (Vaswani et al., 2017) and a deep averaging network (DAN) (Iyyer et al., 2015). These two variants have trade-offs between accuracy and compute resources. We chose the transformer model because it performs better than the DAN model on various NLP tasks (Cer et al., 2018). CCNN is a combination of character-level features trained on temporal (1D) convolutional networks [ConvNets; Zhang et al. (2015)]. It treats input characters in text as a raw-signal which is then applied to ConvNets. Each character in text is encoded using a one-hot vector such that the maximum length l of a character sequence does not exceed a preset length l_0.

4 CLASSIFIER DESIGN

The architecture of our proposed classifier is shown in **Figure 1**. An abstract representation previously discussed is passed to the neural network for encoding. Then the label of the abstract is determined by the output of the sigmoid function that aggregates all word encodings. Note that this architecture is not applicable for use by CCNN or USE. For comparison, we used these two architectures directly as described from their original publications.

LSTM is known for handling the vanishing gradient that occurs when training recurrent neural networks. A typical LSTM cell consists of three gates: input gate $\mathbf{i}_t$, output gate $\mathbf{o}_t$ and forget gate $\mathbf{f}_t$. The input gate updates the cell state; the output gate decides the next hidden state, and the forget gate decides whether to store or erase particular information in the current state $\mathbf{h}_t$. We use $\tanh(\cdot)$ as the activation function and the sigmoid function $\sigma(\cdot)$ to map the output values into a probability distribution. The current hidden state $\mathbf{h}_t$ of LSTM cells can be implemented with the following equations:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{4}$$

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f\right) \tag{5}$$

$$\mathbf{z}_t = \tanh(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{6}$$

$$\mathbf{c}_t = \mathbf{z}_t \odot \mathbf{i}_t + \mathbf{c}_{t-1} \odot \mathbf{f}_t \tag{7}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \tag{8}$$

$$\mathbf{h}_t = \tanh(\mathbf{c}_t) \odot \mathbf{o}_t \tag{9}$$

At a given time step t, $\mathbf{x}_t$ represents the input vector; $\mathbf{c}_t$ represents cell state vector or memory cell; $\mathbf{z}_t$ is a temporary result. $\mathbf{W}$ and $\mathbf{U}$ are weights for the input gate i, forget gate f, temporary result z, and output gate o.

GRU is similar to LSTM, except that it has only a reset gate $\mathbf{r}_t$ and an update gate $\mathbf{z}_t$. The current hidden state $\mathbf{h}_t$ at a given timestep t can be calculated with:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{10}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{11}$$

$$\tilde{h}_t = \tanh(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \tag{12}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{h}_t \tag{13}$$

with the same defined variables. GRU is less computationally expensive than LSTM and achieves comparable or better performance for many tasks. For a given sequence, we train LSTM and GRU in two directions (BiLSTM and BiGRU) to predict the label for the current position using both historical and future data, which has been shown to outperform a single direction model for many tasks.

Attention Mechanism The attention mechanism is used to weight word tokens deferentially when aggregating them into a document level representations. In our system (**Figure 1**), embeddings of words are concatenated into a vector with D_{WE} dimensions. Using the attention mechanism, each word t contributes to the sentence vector, which is characterized by the factor α_t such that

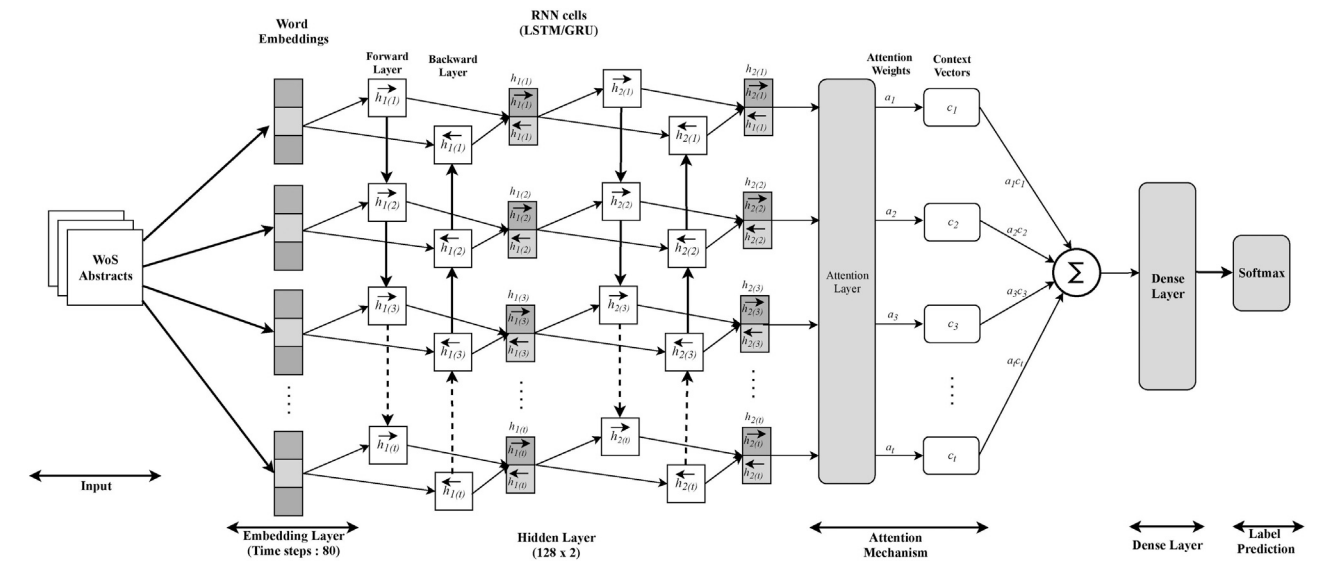

FIGURE 1 | Subject category (SC) classification architecture.

$$\alpha_t = \frac{\exp\left(\mathbf{u}_t^\top \mathbf{v}_t\right)}{\sum_t \exp\left(\mathbf{u}_t^\top \mathbf{v}_t,\right)} \tag{14}$$

$$\mathbf{u}_t = \tanh\left(\mathbf{W} \cdot \mathbf{h}_t + \mathbf{b}\right) \tag{15}$$

in which $\mathbf{h}_t = [\overrightarrow{h}_t; \overleftarrow{h}_t]$ is the representation of each word after the BiLSTM or BiGRU layers, $\mathbf{v}_t$ is the context vector that is randomly initialized and learned during the training process, $\mathbf{W}$ is the weight, and $\mathbf{b}$ is the bias. An abstract vector $\mathbf{v}$ is generated by aggregating word vectors using weights learned by the attention mechanism. We then calculate the weighted sum of $\mathbf{h}_t$ using the attention weights by:

$$\mathbf{v} = \sum_t \alpha_t \mathbf{h}_t. \tag{16}$$

5 EXPERIMENTS

Our training dataset is from the WoS database for the year 2015. The entire dataset contains approximately 45 million records of academic documents, most having titles and abstracts. They are labeled with 235 SCs at the journal level in three broad categories–Science, Social Science, and Art and Literature. A portion of the SCs have subcategories, such as "Physics, Condensed Matter," "Physics, Nuclear," and "Physics, Applied." Here, we collapse these subcategories, which reduces the total number of SCs to 115. We do this because the minor classes decrease the performance of the model (due to the less availability of that data). Also, we need to have an "others" class to balance the data samples. We also exclude papers labeled with more than one category and papers that are labeled as "Multidisciplinary." Abstracts with less than 10 words are excluded. The final number of singly labeled abstracts is approximately nine million, in 104 SCs. The sample sizes of these SCs range from 15 (Art) to 734k (Physics) with a median about 86k. We randomly select up to 150k abstracts per SC. This upper limit is based on our preliminary study (Wu et al., 2018). The ratio between the training and testing corpus is 9:1.

The median of word types per abstract is approximately 80–90. As such, we choose the top $d = 80$ elements from $\mathbf{A}_{\text{sorted}}$ to represent the abstract. If $\mathbf{A}_{\text{sorted}}$ has less than d elements, we pad the feature list with zeros. The word vector dimensions of GloVe and FastText are set to 50 and 100, respectively. This falls into the reasonable value range (24–256) for WE dimensions (Witt and Seifert, 2017). When training the BiLSM and BiGRU models, each layer contains 128 neurons. We investigate the dependency of classification performance on these hyper-parameters by varying the number of layers and neurons. We varied the number of word types per abstract d and set the dropout rate to 20% to mitigate overfitting or underfitting. Due to their relatively large size, we train the neural networks using mini-batch gradient descent with *Adam* for gradient optimization and one word cross entropy as the loss function. The learning rate was set to 10^{-3}.

6 EVALUATION AND COMPARISON

6.1 One-Level Classifier

We first classify all abstracts in the testing set into 104 SCs using the retrained GloVe WE model with BiGRU. The model achieves a micro-F_1 score of 0.71. The first panel in **Figure 2** shows the SCs that achieve the highest F_1's; the second panel shows SCs that achieve relatively low F_1's. The results indicate that the classifier performs poorer on SCs with relatively small sample sizes than SCs with relatively large sample sizes. The data imbalance is likely to contribute to the significantly different performances of SCs.

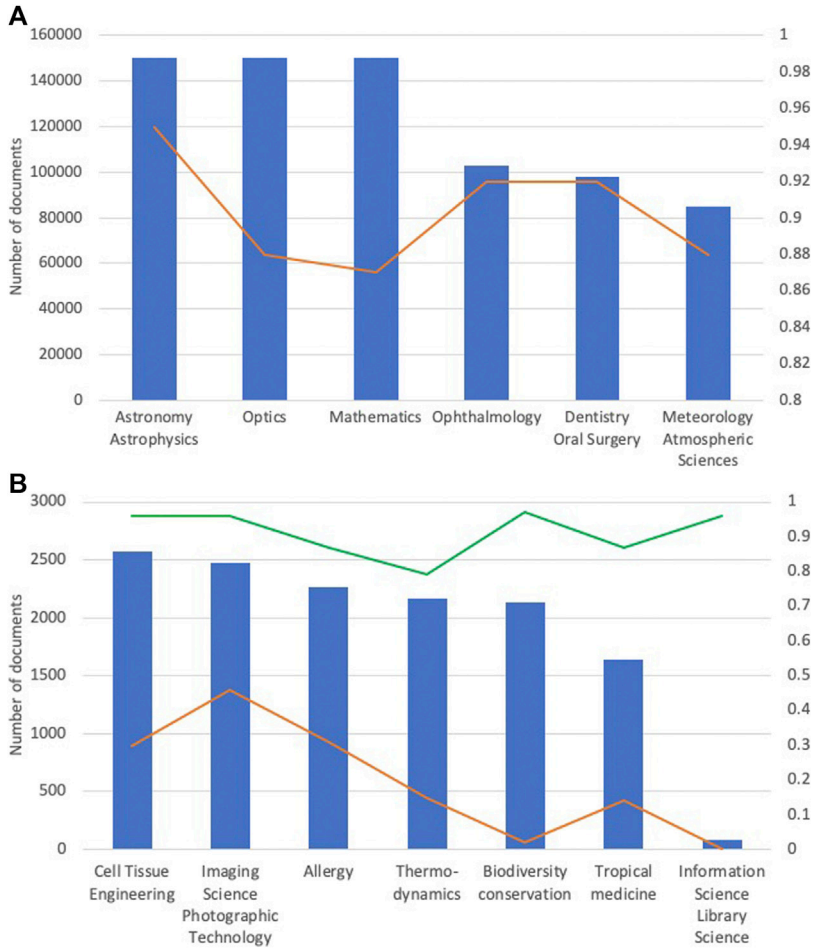

FIGURE 2 | Number of training documents (blue bars) and the corresponding F_1 values (red curves) for best performance **(top)** and worst performance **(bottom)** SC's. Green line shows improved F_1's produced by the second-level classifier.

6.2 Two-Level Classifier

To mitigate the data imbalance problems for the one-level classifier, we train a two-level classifier. The first level classifies abstracts into 81 SCs, including 80 major SCs and an "Others" category, which incorporates 24 minor SCs. "Others" contains the categories with training data <10k abstracts. Abstracts that fall into the "Others" are further classified by a second level classifier, which is trained on abstracts belonging to the 24 minor SCs.

6.3 Baseline Methods

For comparison, we trained five supervised machine learning models as baselines. They are Random Forest (RF), Naïve Bayes (NB, Gaussian), Support Vector Machine (SVM, linear and Radial Basis Function kernels), and Logistic Regression (LR). Documents are represented in the same way as for the DANN except that no word embedding is performed. Because it takes an extremely long time to train these models using all data used for training DANN, and the implementation does not support batch processing, we downsize the training corpus to 150k in total and keep training samples in each SC in proportion to those used in DANN. The performance metrics are calculated based on the same testing corpus as the DANN model.

We used the CCNN architecture (Zhang et al., 2015), which contains six convolutional layers each including 1,008 neurons followed by three fully connected layers. Each abstract is represented by a 1,014 dimensional vector. Our architecture for USE (Cer et al., 2018) is an MLP with four layers, each of which contains 1,024 neurons. Each abstract is represented by a 512 dimensional vector.

6.4 Results

The performances of DANN in different settings and a comparison between the best DANN models and baseline models are illustrated in **Figure 3**. The numerical values of performance metrics using the two-level classifier are tabulated in **Supplementary Table S1**. Below are the observations from results.

(1) FastText + BiGRU + Attn and FastText+BiLSTM + Attn achieve the highest micro-F_1 of 0.76. Several models achieve similar results:GloVe + BiLSTM + Attn (micro-F_1 = 0.75),

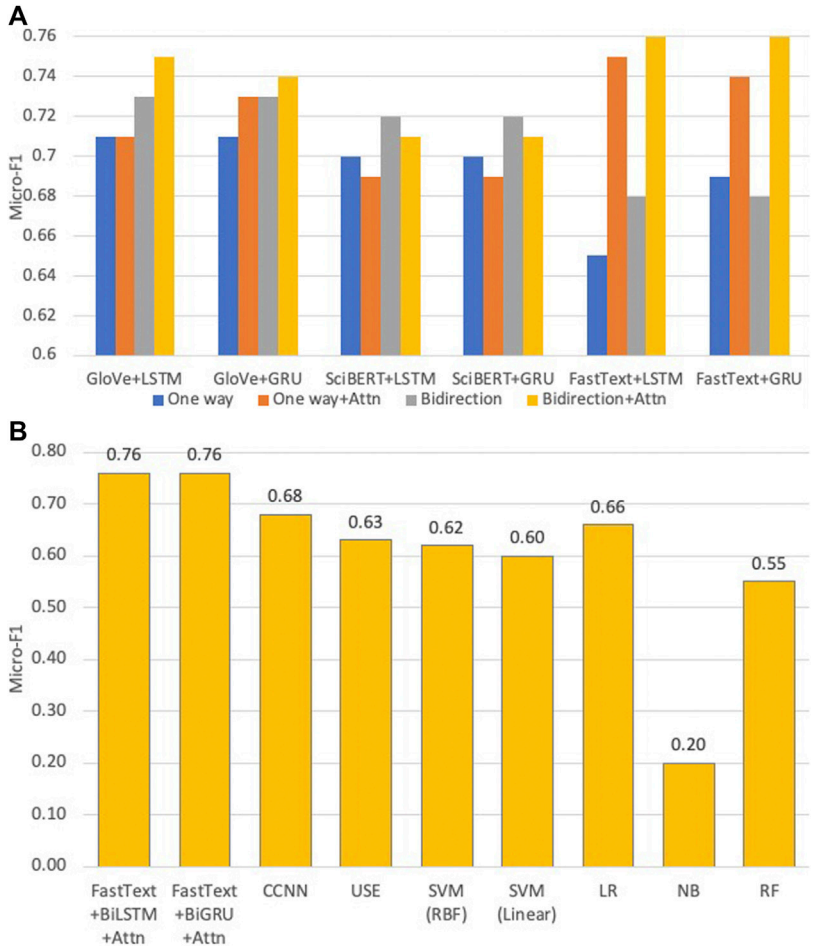

FIGURE 3 | Top: Micro-F_1's of our DANN models that classify abstracts into 81 SCs. Variants of models within each group are color-coded. Bottom: Micro-F_1's of our best DANN models that classify abstracts into 81 SCs, compared with baseline models.

GloVe + BiGRU + Attn (micro-F_1 = 0.74), FastText + LSTM + Attn (micro-F_1 = 0.75), and FastText + GRU + Attn (micro-F_1 = 0.74). These results indicate that the attention mechanism significantly improves the classifier performance.

(2) Retraining FastText and GloVe significantly boosted the performance. In contrast, the best micro-F_1 achieved by USE is 0.64, which is likely resulted from its relatively low vocabulary overlap. Another reason could be is that the single vector of fixed length only encodes the overall semantics of the abstract. The occurrences of words are better indicators of sentences in specific domains.

(3) LSTM and GRU and their bidirectional counterparts exhibit very similar performance, which is consistent with a recent systematic survey (Greff et al., 2017).

(4) For FastText + BiGRU + Attn, the F_1 measure varies from 0.50 to 0.95 with a median of 0.76. The distribution of F_1 values for 81 SCs is shown in **Figure 4**. The F_1 achieved by the first-level classifier with 81 categories (micro-F_1 = 0.76) is improved compared with the classifier trained on 104 SCs (micro-F_1 = 0.70)

(5) The performance was not improved by increasing the GloVe vector dimension from 50 to 100 (not shown) under the setting of GloVe + BiGRU with 128 neurons on two layers which is consistent with earlier work (Witt and Seifert, 2017).

(6) Word-level embedding models in general perform better than the character-level embedding models (i.e., CCNN). CCNN considers the text as a raw-signal, so the word vectors constructed are more appropriate when comparing morphological similarities. However, semantically similar words may not be morphologically similar, e.g., "Neural Networks" and "Deep Learning."

(7) SciBERT's performance is 3–5% below FastText and GloVe, indicating that re-trained WE models exhibit an advantage over pre-trained WE models. This is because SciBERT was trained on the PubMed corpus which mostly incorporates papers in biomedical and life sciences. Also, due to their large dimensions, the training time was greater than FastText under the same parameter settings.

(8) The best DANN model beats the best machine learning model (LR) by about 10%.

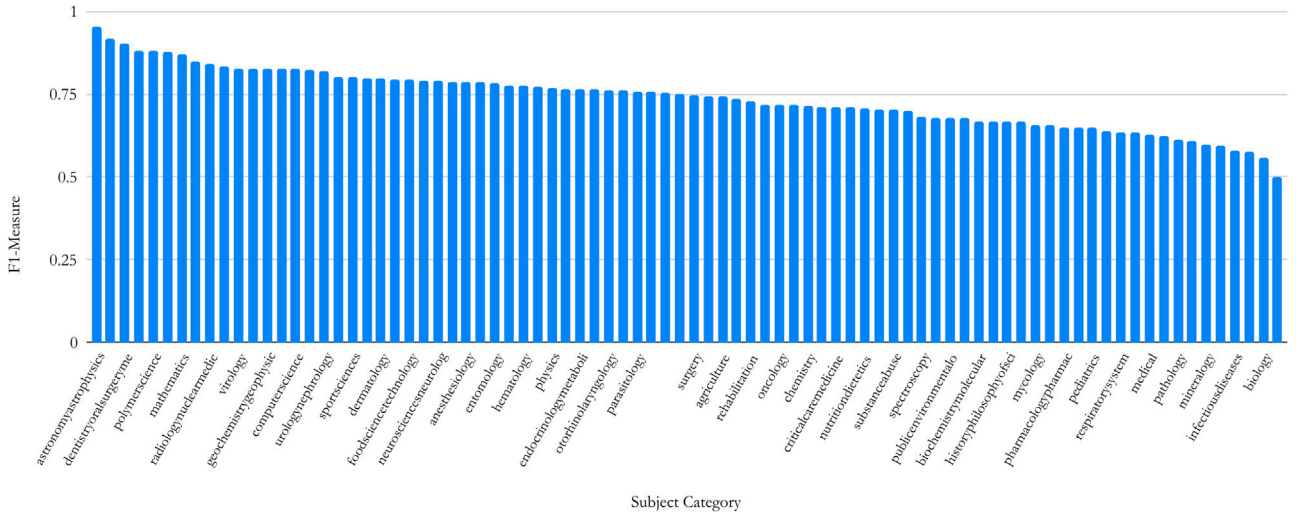

FIGURE 4 | Distribution of F_1's across 81 SC's obtained by the first level classifier.

We also investigated the dependency of classification performance on key hyper-parameters. The settings of GLoVe + BiGRU with 128 neurons on two layers are considered as the "reference setting." With the setting of GloVe + BiGRU, we increase the neuron number by factor of 10 (1,280 neurons on two layers) and obtained marginally improved performance by 1% compared with the same setting with 128 neurons. We also doubled the number of layers (128 neurons on four layers). Without attention, the model performs worse than the reference setting by 3%. With the attention mechanism, the micro-F_1 = 0.75 is marginally improved by 1% with respect to the reference setting. We also increase the default number of neurons of USE to 2048 neurons for four layers. The micro-F_1 improves marginally by 1%, reaching only 0.64. The results indicate that adding more neurons and layers seem to have little impact to the performance improvement.

The second-level classifier is trained using the same neural architecture as the first-level on the "Others" corpus. **Figure 2** (Right ordinate legend) shows that F_1's vary from 0.92 to 0.97 with a median of 0.96. The results are significantly improved by classifying minor classes separately from major classes.

TABLE 1 | Results of the top 10 SCs of classifying one million research papers in CiteSeerX, using our best model.

Rank	Subject categories	Fraction (%)
1	Biology	23.85
2	Computer science	19.17
3	Mathematics	5.06
4	Engineering	4.97
5	Public environment	3.45
6	Physics	3.16
7	Environmental sciences	1.81
8	Astronomy astrophysics	1.79
9	Neurosciences neurology	1.52
10	Chemistry	1.47

7 DISCUSSION

7.1 Sampling Strategies

The data imbalance problem is ubiquitous in both multi-class and multi-label classification problems (Charte et al., 2015). The imbalance ratio (IR), defined as the ratio of the number of instances in the majority class to the number of samples in the minority class (García et al., 2012), has been commonly used to characterize the level of imbalance. Compared with the imbalance datasets in **Table 1** of (Charte et al., 2015), our data has a significantly high level of imbalance. In particular, the highest IR is about 49,000 (#Physics/#Art). One commonly used way to mitigate this problem is data resampling. This method is based on rebalancing SC distributions by either deleting instances of major SCs (undersampling) or supplementing artificially generated instances of the minor SCs (oversampling). We can always undersample major SCs, but this means we have to reduce sample sizes of all SCs down to about 15 (Art; **Section 5**), which is too small for training robust neural network models. The oversampling strategies such as SMOTE (Chawla et al., 2002) works for problems involving continuous numerical quantities, e.g., SalahEldeen and Nelson (2015). In our case, the synthesized vectors of "abstracts" by SMOTE will not map to any actual words because word representations are very sparsely distributed in the large WE space. Even if we oversample minor SCs using semantically dummy vectors, generating all samples will take a large amount of time given the high dimensionality of abstract vectors and high IR. Therefore, we only use real data.

7.2 Category Overlapping

We discuss the potential impact on classification results contributed by categories overlapping in the training data. Our

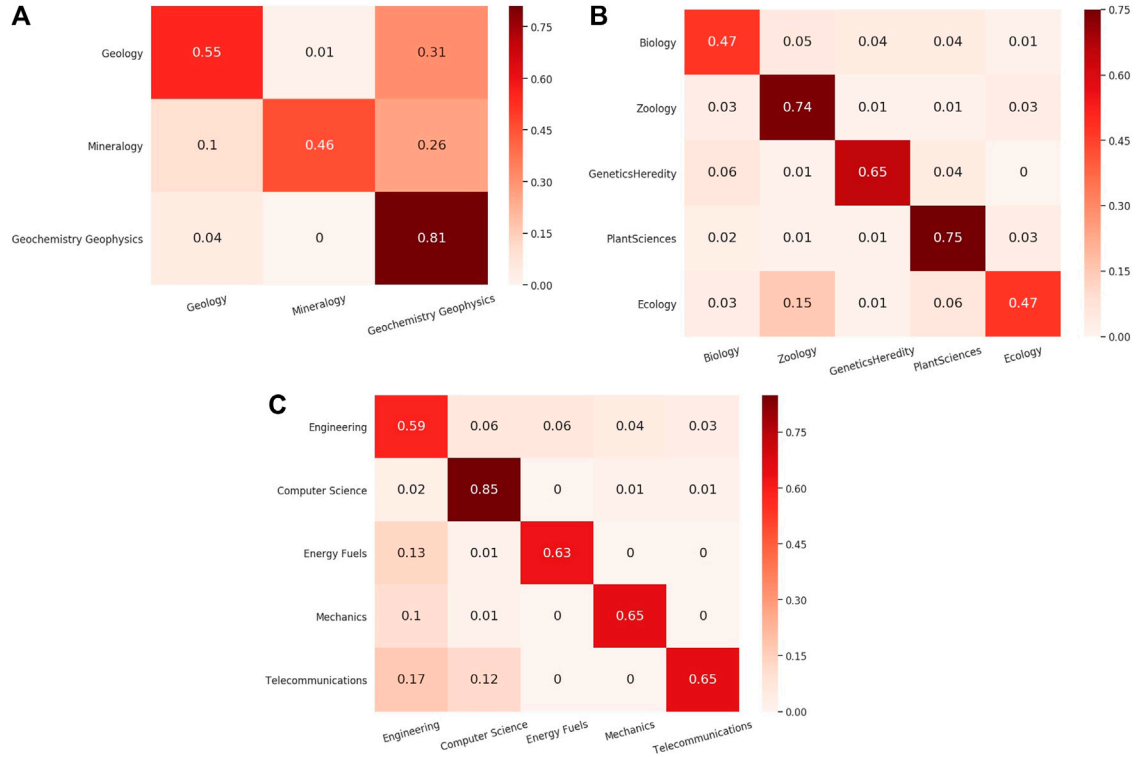

FIGURE 5 | Normalized Confusion Matrix for closely related classes in which a large fraction of "Geology" and "Mineralogy" papers are classified into "GeoChemistry GeoPhysics" **(A)**, and a large fraction of Zoology papers are classified into "biology" or "ecology" **(B)**, a large fraction of "TeleCommunications," "Mechanics" and "EnergyFuels" papers are classified into "Engineering" **(C)**.

initial classification schema contains 104 SCs, but they are not all mutually exclusive. Instead, the vocabularies of some categories overlap with the others. For example, papers exclusively labeled as "Materials Science" and "Metallurgy" exhibit significant overlap in their tokens. In the WE vector space, the semantic vectors labeled with either category are overlapped making it hard to differentiate them. **Figure 5** shows the confusion matrices of the closely related categories such as "Geology," "Mineralogy," and "Geochemistry Geophysics." **Figure 6** is the *t*-SNE plot of abstracts of closely related SCs. To make the plot less crowded, we randomly select 250 abstracts from each SC as shown in **Figure 5**. Data points representing "Geology," "Mineralogy," and "Geochemistry Geophysics" tend to spread or are overlapped in such a way that are hard to be visually distinguished.

One way to mitigate this problem is to merge overlapped categories. However, special care should be taken on whether these overlapped SCs are truly strongly related and should be evaluated by domain experts. For example, "Zoology," "PlantSciences," and "Ecology" can be merged into a single SC called "Biology" (Gaff, 2019; private communication). "Geology," "Mineralogy," and "GeoChemistry GeoPhysics" can be merged into a single SC called "Geology." However, "Materials Science" and "Metallurgy" may not be merged (Liu, 2019; private communication) to a single SC. By doing the aforementioned merges, the number of SCs is reduced to 74. As a preliminary study, we classified the merged dataset using our best model (retrained FastText + BiGRU + Attn) and achieved an improvement with an overall micro-F_1 score of 0.78. The classification performance of "Geology" after merging has improved from 0.83 to 0.88.

7.3 Limitations

Compared with existing work, our models are trained on a relatively comprehensive, large-scale, and clean dataset from WoS. However, the basic classification of WoS is at the journal level and not at the article level. We are also aware that the classification schema of WoS may change over time. For example, in 2018, WoS introduced three new SCs such as Quantum Science and Technology, reflecting emerging research trends and technologies (Boletta, 2019). To mitigate this effect, we excluded papers with multiple SCs and assume that the SCs of papers studied are stationary and journal level classifications represent the paper level SCs.

Another limitation is the document representation. The BoW model ignores the sequential information. Although we experimented on the cases in which we keep word tokens in the same order as they appear in the original documents, the exclusion of stop words breaks the original sequence, which is the input of the recurrent encoder. We will address this limitation in

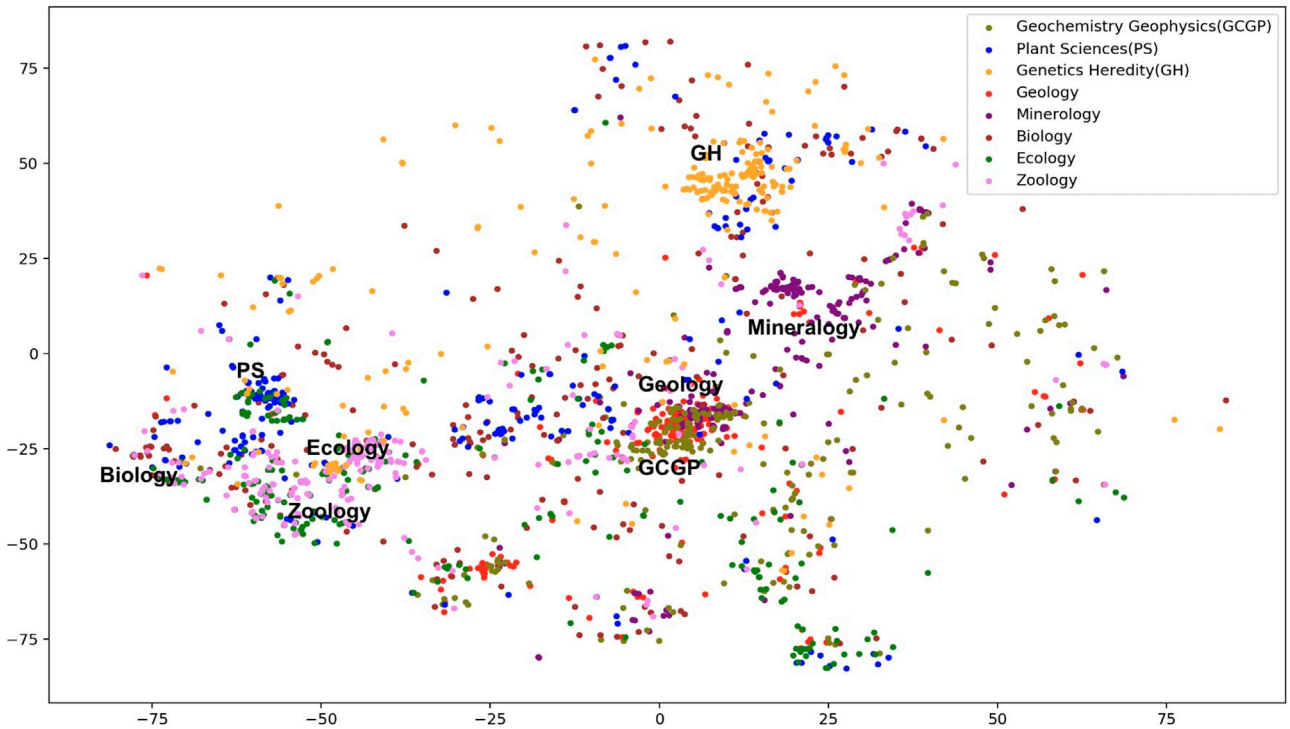

FIGURE 6 | *t*-SNE plot of closely related SCs.

future research by encoding the whole sentences, e.g., Yang et al. (2016).

8 APPLICATION TO CITESEERX

CiteSeerX is a digital library search engine that was the first to use automatic citation indexing (Giles et al., 1998). It is an open source search engine that provides metadata and full-text access for more than 10 million scholarly documents and continues to add new documents (Wu et al., 2019). In the past decade, it has incorporated scholarly documents in diverse SCs, but the distribution of their subject categories is unknown. Using the best neural network model in this work (FreeText + BiGRU + Attn), we classified one million papers randomly selected from CiteSeerX into 104 SCs (**Table 1**). The fraction of Computer Science papers (19.2%) is significantly higher than the results in Wu et al. (2018), which was 7.58%. The F_1 for Computer Science was about 0.94 for Computer Science which is higher than this work (about 0.80). Therefore, the fraction may be overestimated here. However Wu et al. (2018), had only six classes and this model classifies abstracts into 104 SCs, so although this compromises the accuracy (by around 7% on average), our work can still be used as a starting point for a systematic SC classification. The classifier classifies one million abstracts in 1,253 s implying that will be scalable on multi-millions of papers.

9 CONCLUSION

We investigated the problem of systematically classifying a large collection of scholarly papers into 104 SC's using neural network methods based only on abstracts. Our methods appear to scale better than existing clustering-based methods relying on citation networks. For neural network methods, our retrained FastText or GloVe combined with BiGRU or BiLSTM with the attention mechanism gives the best results. Retraining WE models and using an attention mechanism play important roles in improving the classifier performance. A two-level classifier effectively improves our performance when dealing with training data that has extremely imbalanced categories. The median F_1's under the best settings are 0.75–0.76.

One bottleneck of our classifier is the overlapping categories. Merging closely related SCs is a promising solution, but should be under the guidance of domain experts. The TF-IDF representation only considers unigrams. Future work could consider n-grams or concepts ($n \geq 2$) and transfer learning to adopt word/sentence embedding models trained on non-scholarly corpora (Arora et al., 2017; Conneau et al., 2017). One could investigate models that also take into account stop-words, e.g., Yang et al. (2016). One could also explore alternative optimizers of neural networks besides Adam, such as the Stochastic Gradient Descent (SGD). Our work falls into the multiclass classification, which classifies research papers into flat SCs. In the future, we will investigate hierarchical

multilabel classification that assigns multiple SCs at multiple levels to papers.

DATA AVAILABILITY STATEMENT

The Web of Science (WoS) dataset used for this study is proprietary and can be purchased from Clarivate[6]. The implementation software is open accessible from GitHub[7]. The testing datasets and CiteSeerX classification results are available on figshare[8].

AUTHOR CONTRIBUTIONS

BK designed the study and implemented the models. He is responsible for analyzing the results and writing the paper. SR is responsible for model selection, experiment design, and reviewing methods and results. JW was responsible for data management, selection, and curation. He reviewed related works and contributed to the introduction. CG is responsible for project management, editing, and supervision.

FUNDING

This research is partially funded by the National Science Foundation (Grant No: 1823288).

ACKNOWLEDGMENTS

We gratefully acknowledge partial support from the National Science Foundation. We also acknowledge Adam T. McMillen for technical support, and Holly Gaff, Old Dominion University and Shimin Liu, Pennsylvania State University as domain experts respectively in biology and the earth and mineral sciences.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frma.2020.600382/full#supplementary-material.

REFERENCES

Arora, S., Liang, Y., and Ma, T. (2017). "A simple but tough-to-beat baseline for sentence embeddings," in ICLR, Toulon, France, April 24-26, 2017.

Barnes, B., Bloor, D., and Henry, J. (1996). *Scientific knowledge: a sociological analysis*. Chicago IL: University of Chicago Press.

Beltagy, I., Cohan, A., and Lo, K. (2019). Scibert: pretrained contextualized embeddings for scientific text. arXiv:1903.10676.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Trans. Assoc. Comput.* 5, 135–146. doi:10.1162/tacl_a_00051

Boletta, M. (2019). New web of science categories reflect ever-evolving research. Available at: https://clarivate.com/webofsciencegroup/article/new-web-of-science-categories-reflect-ever-evolving-research/ (Accessed January 24, 2019).

Boyack, K. W., and Klavans, R. (2018). *Accurately identifying topics using text: mapping pubmed*. Leiden, Netherlands: Centre for Science and Technology Studies (CWTS), 107–115.

Caragea, C., Wu, J., Gollapalli, S. D., and Giles, C. L. (2016). "Document type classification in online digital libraries," in Proceedings of the 13th AAAI conference, Phoenix, AZ, USA, February 12-17, 2016.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., et al. (2018). "Universal sentence encoder for English," in Proceedings of EMNLP conference, Brussels, Belgium, October 31-November 4, 2018.

Charte, F., Rivera, A. J., del Jesús, M. J., and Herrera, F. (2015). Addressing imbalance in multilabel classification: measures and random resampling algorithms. *Neurocomputing* 163, 3–16. doi:10.1016/j.neucom.2014.08.091

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Jair* 16, 321–357. doi:10.1613/jair.953

Cho, K., Van Merrienboer, B., Gülçehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al.(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 2014, Doha Qatar, October 25-29, 2014, 1724–1734.

Collins, H. M. (1992). "Epistemological chicken hm collins and steven yearley," in *Science as practice and culture*. Editor A. Pickering (Chicago, IL: University of Chicago Press), 301.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). "Supervised learning of universal sentence representations from natural language inference data," in Proceedings of the EMNLP conference, Copenhagen, Denmark, September 9-11, 2017.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). "BERT: pre-training of deep bidirectional transformers for language understanding," in NAACL-HLT, Minneapolis, MN, USA, June 2-7, 2019.

Fellbaum, C. (2005). "Wordnet and wordnets," in *Encyclopedia of language and linguistics*. Editor A. Barber (Elsevier), 2–665.

Fortunato, S., Bergstrom, C. T., Börner, K., Evans, J. A., Helbing, D., Milojević, S., et al. (2018). Science of science. *Science* 359, eaao0185. doi:10.1126/science.aao0185

García, V., Sánchez, J. S., and Mollineda, R. A. (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowl. Base Syst.* 25, 13–21. doi:10.1016/j.knosys.2011.06.013

Gerlach, M., Peixoto, T. P., and Altmann, E. G. (2018). A network approach to topic models. *Sci. Adv.* 4, eaaq1360. doi:10.1126/sciadv.aaq1360

Giles, C. L., Bollacker, K. D., and Lawrence, S. (1998). "CiteSeer: An automatic citation indexing system," in Proceedings of the 3rd ACM international conference on digital libraries, June 23–26, 1998, Pittsburgh, PA, United States, 89–98.

Glänzel, W., and Thijs, B. (2017). Using hybrid methods and 'core documents' for the representation of clusters and topics: the astronomy dataset. *Scientometrics* 111, 1071–1087. doi:10.1007/s11192-017-2301-6

Goldberg, Y., and Levy, O. (2014). Word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. preprint arXiv:1402.3722.

Grave, E., Mikolov, T., Joulin, A., and Bojanowski, P. (2017). "Bag of tricks for efficient text classification," in Proceedings of the 15th EACL, Valencia, Span, April 3-7, 2017.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2017). LSTM: a search space odyssey. *IEEE Trans. Neural Networks Learn. Syst.* 28, 2222–2232. doi:10.1109/TNNLS.2016.2582924

He, G., Fang, J., Cui, H., Wu, C., and Lu, W. (2018). "Keyphrase extraction based on prior knowledge," in Proceedings of the 18th ACM/IEEE on joint conference on digital libraries, JCDL, Fort Worth, TX, USA, June 3-7, 2018.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780.

Iyyer, M., Manjunatha, V., Boyd-Graber, J., and Daumé, H., III (2015). "Deep unordered composition rivals syntactic methods for text classification," in Proceedings ACL, Beijing, China, July 26-31, 2015.

[6]https://clarivate.libguides.com/rawdata

[7]https://github.com/SeerLabs/sbdsubjectclassifier

[8]https://doi.org/10.6084/m9.figshare.12887966.v2

Khabsa, M., and Giles, C. L. (2014). The number of scholarly documents on the public web. *PloS One* 9, e93949. doi:10.1371/journal.pone.0093949

Larsen, P., and von Ins, M. (2010). The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics* 84, 575–603. doi:10.1007/s11192-010-0202-z

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., et al. (1989). "Handwritten digit recognition with a back-propagation network," in Advances in neural information processing systems [NIPS conference], Denver, Colorado, USA, November 27-30, 1989.

Li, Y., Lipsky Gorman, S., and Elhadad, N. (2010). "Section classification in clinical notes using supervised hidden markov model," in Proceedings of the 1st ACM international health informatics symposium, Arlington, VA, USA, November 11-12, 2010 (New York, NY: Association for Computing Machinery), 744–750.

Matsuda, K., and Fukushima, T. (1999). Task-oriented world wide web retrieval by document type classification. In Proceedings of CIKM, Kansas City, Missouri, USA, November 2-6, 1999.

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

Mikolov, T., Yih, W., and Zweig, G. (2013b). "Linguistic regularities in continuous space word representations," in Proceedings of NAACL-HLT, Atlanta, GA, USA, June 9-14, 2013.

Moscato, P., and Cotta, C. (2003). *A gentle introduction to memetic algorithms*. Boston, MA: Springer US, 105–144.

Nadeau, D., and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticæ Investigationes* 30, 3–26. doi:10.1075/li.30.1.03nad

Nam, S., Jeong, S., Kim, S.-K., Kim, H.-G., Ngo, V., and Zong, N. (2016). Structuralizing biomedical abstracts with discriminative linguistic features. *Comput. Biol. Med.* 79, 276–285. doi:10.1016/j.compbiomed.2016.10.026

Passos, A., Kumar, V., and McCallum, A. (2014). "Lexicon infused phrase embeddings for named entity resolution," in Proceedings of CoNLL, Baltimore, MD, USA, June, 26-27, 2014.

Pennington, J., Socher, R., and Manning, C. D. (2014). "Glove: global vectors for word representation," in Proceedings of the EMNLP conference, Doha, Qatar, October 25-29, 2014.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). "Deep contextualized word representations," in NAACL-HLT, New Orleans, LA,, USA, June 1-6, 2018.

Prasad, A., Kaur, M., and Kan, M.-Y. (2018). Neural ParsCit: a deep learning-based reference string parser. *Int. J. Digit. Libr.* 19, 323–337. doi:10.1007/2Fs00799-018-0242-1

Ratnaparkhi, A. (1996). "A maximum entropy model for part-of-speech tagging," in The proceedings of the EMNLP conference, Philadelphia, PA, USA, May 17-18, 1996.

Ren, Y., Zhang, Y., Zhang, M., and Ji, D. (2016). "Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings," in AAAI, Phoenix, AZ, USA, February 12-17, 2016.

SalahEldeen, H. M., and Nelson, M. L. (2015). "Predicting temporal intention in resource sharing," in Proceedings of the 15th JCDL conference, Knoxville, TN, USA, June 21-25, 2015.

Shen, Z., Ma, H., and Wang, K. (2018). "A web-scale system for scientific knowledge exploration," in Proceedings of ACL 2018, system demonstrations (Melbourne, Australia: Association for Computational Linguistics), 87–92.

van Eck, N. J., and Waltman, L. (2017). Citation-based clustering of publications using citnetexplorer and vosviewer. *Scientometrics* 111, 1053–1070. doi:10.1007/s11192-017-2300-7

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). "Attention is all you need," in NIPS, Long Beach, CA, USA, December 4-9, 2017.

Vo, D.-T., and Zhang, Y. (2015). "Target-dependent twitter sentiment classification with rich automatic features," In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015, Editors Q. Yang and M. J. Wooldridge (AAAI Press), 1347–1353.

Waltman, L., and van Eck, N. J. (2012). A new methodology for constructing a publication-level classification system of science. *JASIST* 63, 2378–2392. doi:10.1002/asi.22748

Wang, S., and Koopman, R. (2017). Clustering articles based on semantic similarity. *Scientometrics* 111, 1017–1031. doi:10.1007/s11192-017-2298-x

Witt, N., and Seifert, C. (2017). "Understanding the influence of hyperparameters on text embeddings for text classification tasks," in TPDL conference, Thessaloniki, Greece, September 18-21, 2017.

Wu, J., Kandimalla, B., Rohatgi, S., Sefid, A., Mao, J., and Giles, C. L. (2018). "Citeseerx-2018: a cleansed multidisciplinary scholarly big dataset," in IEEE big data, Seattle, WA, USA, December 10-13, 2018.

Wu, J., Kim, K., and Giles, C. L. (2019). "CiteSeerX: 20 years of service to scholarly big data," in Proceedings of the AIDR conference, Pittsburgh, PA, USA, May 13-15, 2019.

Wu, J., Williams, K., Chen, H., Khabsa, M., Caragea, C., Ororbia, A., et al. (2014). "CiteSeerX: AI in a digital library search engine," in Proceedings of the twenty-eighth AAAI conference on artificial intelligence.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J., and Hovy, E. H. (2016). "Hierarchical attention networks for document classification," in The NAACL-HLT conference.

Zhang, X., Zhao, J., and LeCun, Y. (2015). "Character-level convolutional networks for text classification," in Proceedings of the NIPS conference, Montreal, Canada, December 7-12, 2015.

Zitt, M., Lelu, A., Cadot, M., and Cabanac, G. (2019). "Bibliometric delineation of scientific fields," in *Handbook of science and technology indicators*. Editors W. Glänzel, H. F. Moed, U. Schmoch, and M. Thelwall (Springer International Publishing), 25–68.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Kandimalla, Rohatgi, Wu and Giles. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.