Knowledge and Information Systems
https://doi.org/10.1007/s10115-023-01856-z

REVIEW

Text classification using embeddings: a survey

Liliane Soares da Costa[1] · **Italo L. Oliveira**[1] · **Renato Fileto**[1]

Received: 15 April 2022 / Revised: 2 July 2022 / Accepted: 9 March 2023

Abstract
Text classification results can be hindered when just the bag-of-words model is used for representing features, because it ignores word order and senses, which can vary with the context. Embeddings have recently emerged as a means to circumvent these limitations, allowing considerable performance gains. However, determining the best combinations of classification techniques and embeddings for classifying particular corpora can be challenging. This survey provides a comprehensive review of text classification approaches that employ embeddings. First, it analyzes past and recent advancements in feature representation for text classification. Then, it identifies the combinations of embedding-based feature representations and classification techniques that have provided the best performances for classifying text from distinct corpora, also providing links to the original articles, source code (when available) and data sets used in the performance evaluation. Finally, it discusses current challenges and promising directions for text classification research, such as cost-effectiveness, multi-label classification, and the potential of knowledge graphs and knowledge embeddings to enhance text classification.

Keywords Text classification · Feature representation · Embeddings

1 Introduction

The significant and continuous increase in the volume of digital data gained much attention in recent years. An EMC study [1], with data collected between 2005 and 2020, showed how the "digital universe" expands, doubling the amount of data every two years. In this

Italo L. Oliveira and Renato Fileto have equally contributed to this work.

✉ Liliane Soares da Costa
liliane.costa@posgrad.ufsc.br

Italo L. Oliveira
italo.oliveira@posgrad.ufsc.br

Renato Fileto
r.fileto@ufsc.br

[1] Department of Informatics and Statistics (INE), Federal University of Santa Catarina (UFSC), Campus Reitor João David Ferreira Lima, Florianópolis, SC 88040-900, Brazil

Published online: 26 March 2023

universe of data, text documents are produced in large amounts and can be rich in information and knowledge. Huge volumes of text have been gathered continuously by online libraries, websites, and social media, among other sources. Textual data tend to continue increasing with new developments in technology such as greater cloud storage capacity, speech-to-text tools, and digital assistants. Therefore, processing textual contents automatically is essential to take advantage of them in many applications. However, text processing can be problematic. To tackle this problem, natural language processing (NLP), machine learning (ML), and data mining (DM) techniques work mutually to perform different tasks and extract information from a variety of documents. Text classification [2] is a crucial task in this scenario. It became a crucial means to help handle large collections of documents and has been successfully applied to problems like spam filtering [3, 4], sentiment analysis [5, 6], and opinion detection [7].

The text classification task assigns tags or categories to text documents according with their contents [8]. Formally, given a set of documents D and a set of predefined categories C, the problem of text classification can be modeled as finding a mapping function F from the Cartesian product $D \times C$ to a set $\{True, False\}$, i.e., $F : D \times C \rightarrow \{True, False\}$. Such a mapping function F is called a classifier. For example, based on this mapping, for a document $d_i \in D$ and a category $c_j \in C$, if $F(d_i, c_j) = True$, then d_i belongs to category c_j, otherwise d_i does not belong to c_j [9]. Notice that this definition allows the classification to be multi-class ($|C| > 2$) and multi-label [10] (at least one document belonging to more than one label), though it can be restricted to binary and single label classification.

Nowadays, most of the state-of-the-art text classification methods are based on machine learning or deep learning techniques. Since the inputs of these techniques are usually fixed-length feature vectors, documents are often represented in a bag-of-words (BoW) vector space model. In such a model, each dimension corresponds to one word, and the dimensionality of each vector is the size of the vocabulary. Thus, the vectors are usually high dimensional and sparse. In addition, BoW ignores word order, word senses (which can vary according with the context), and their semantic relations.

Word embeddings have been employed to overcome these limitations, rendering considerable performance gains in various NLP tasks, including text classification. They also represent each word as a vector in a multidimensional space [11]. More formally, given a set of words W, we can define a word embedding as a parameterized function $Emb : W \rightarrow \mathbb{R}^n$, which maps words to n-dimension real-valued vectors. However, vectors in embedded spaces are usually much more compact than those of a BoW, with typical dimensionalities varying from 50 to 500 dimensions. These vectors represent semantic and syntactic properties of words, such as semantic similarity, in a compact format that is suitable for fast processing. Such properties are captured by training the embedding model with vast amounts of text in which the words appear.

Word embedding techniques are often lumped into the deep learning field. This approach was first presented by Bengio [12], but gained popularity with Word2Vec [13]. Nowadays, there are several other competitive word embedding techniques, like GloVe [14], FastText [15], and BERT [16]. They have been the most prominent models of word embeddings. All of them are unsupervised and take a corpus or a dataset as input to generate the vectors.

There are plenty of surveys on text classification [2, 17–25]. However, they compare the proposals according to the techniques employed, and general types of feature representations employed. The only one with focus on semantics is Altinel and Ganiz 2018 [2], which explores advancements in semantic text classification, i.e., approaches that take into account the meaning of the words and semantic connections between them, mainly meaning similarity and sometimes semantic relations. It highlights the advantages of semantic text classification over traditional text classification and organizes existing approaches under five fundamental

categories, namely domain knowledge-based approaches, corpus-based approaches, deep learning-based approaches, word/character sequence-enhanced approaches, and linguistic-enriched approaches.

The surveys of Agarwal and Zhai [17] and Kowsari et al. [22] provide brief overviews of text classification proposals, covering different text features extracted, dimensionality reduction methods, classification algorithms and techniques, and evaluation methods. Agarwal et al. [19] and Kadhim et al. [21], on the other hand, focus on machine learning techniques for automatic text classification and feature selection techniques for reducing high-dimensional feature vectors.

Both Zhou et al. [23] and Minaee et al. [25] provide a review of deep learning-based models for text classification developed in recent years and discuss their technical contributions, similarities, and strengths. They also discuss how the general methods of deep learning deal with text classification problems. Finally, Stein et al. [26] investigate the application of word embeddings and algorithms on hierarchical text classification by employing experimentation and analysis.

This survey provides a comprehensive review of a variety of text classification approaches that use embeddings to represent features. Besides approaches that employ word embeddings, it covers the ones that employ other kinds of embeddings, like document embeddings and knowledge embeddings. It also considers approaches that employ extra information and/or knowledge from sources like Wikipedia, Knowledge Graphs (KG), and their embeddings to semantically enrich the texts prior to classification, with the intent to improve results, specially when context information is limited, or the classification task requires further and precise semantic descriptions of what is mentioned in the texts. To the best of our knowledge, this is the first survey of text classification approaches that focus on embeddings in general, including the ones trained with external information and knowledge sources, such as wikis, thesaurus, taxonomies, ontologies and KGs. One of the major contributions of this work is providing an overview of the distinct kinds of embeddings and classification techniques that have been employed in text classification approaches, identifying the combinations that have provided the highest performance for distinct corpora. This can help the research community and practitioners to devise suitable classification techniques and features to be employed for distinct text classification enterprises according to the domain, data nature, classification categories, etc.

1.1 Outline

The remaining of this paper is structured as follows: Section 2 reports the bibliographical research method employed to find and select the works discussed in this survey. Section 3 describes and illustrates how feature representations that address semantics can improve text classification, and provide an overview of the kinds of embeddings that have been or can be used in this task. Then, Sect. 4 reviews recent approaches selected from the literature that employ different kinds of embeddings as feature representation. Section 5 provides a comparative analysis of these approaches and discusses challenges and the potential of artifacts such as KGs and knowledge embeddings in the future approaches for text classification. Finally, Sect. 6 concludes the paper and presents promising directions for research.

2 Bibliographical research method

The research method for this survey follows the formal systematic literature review methodology. In particular, this study is based on the guidelines proposed in [27] and [28]. As detailed below, we also took into account other surveys about related topics, such as text classification and embeddings.

2.1 Research questions

The goal of this survey is to provide an analysis of existing text classification approaches that employ embeddings, not just word embeddings, but also approaches that employ other kinds of embeddings or that use word embeddings enriched with other types of data or knowledge. To achieve this goal, we aim to answer the following general research question:

> How can different kinds of embeddings improve distinct approaches for text classification?

This question was then divided into four sub-questions as follows:

1. What are state-of-the-art approaches for text classification which use embeddings?
2. Which embeddings are applied in text classification?
3. How do embeddings contribute to the text classification task in the approaches that use them?
4. Can knowledge from other sources (e.g., a KG) benefit text classification?

2.2 Research strategy

The steps of the process employed in the systematic bibliographical review are presented in Fig. 1. The search for papers was done in Google Scholar,[1] the ACM Digital Library,[2] Science Direct,[3] ACL Anthology [4] and, Scopus.[5] We chose these platforms because they gave the best results in preliminary searches for books and papers published in conference proceedings or journals. The search string used was ("Text Classification" OR "Document Classification") AND "Embeddings". We have noticed that this simple search string provides better results than a complex one, with the drawback that we had to filter more works in the following steps.

We found 38045 articles. To reduce this number, in Step 1 we removed duplicates using Mendeley.[6] Then, we considered only articles that satisfied the following criteria: (i) peer-reviewed; (ii) published from 2015 onwards; (iii) written in English; (iv) text classification approaches that use embeddings. Step 1 yielded 860 articles. In Step 2, we manually analyzed the title and the abstract of the 860 articles resulting from Step 1 and removed those not related to text classification. Lastly, in Step 3, we reviewed the remaining 115 articles and selected the 38 ones that use embeddings.

[1] https://scholar.google.com.br/.

[2] https://dl.acm.org/.

[3] https://www.sciencedirect.com/.

[4] https://aclanthology.org/.

[5] https://www.scopus.com/.

[6] https://www.mendeley.com/.

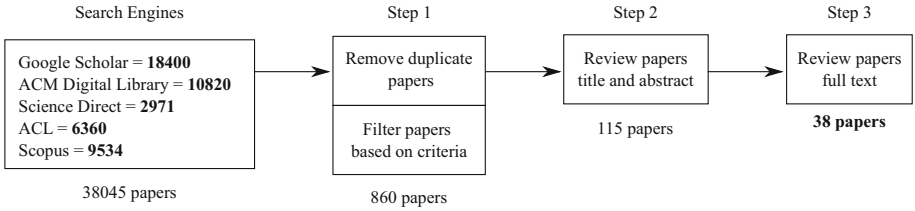


Fig. 1 Process employed to find and select the papers analyzed in this survey

3 Semantics and embeddings in text classification

This section aims to draw how semantics and embeddings can contribute to text classification. First, Sect. 3.1 exemplifies the potential benefits of semantics in text classification in a real-world scenario. Then, Sect. 3.2 describes key aspects of text classification approaches that use embeddings and establish fundamental criteria to identify and analyze these approaches. Finally, Sect. 3.3 gives an overview of the types of embeddings that can be used as features for text classification.

3.1 Motivating example

Semantics is crucial for text classification in certain circumstances. Semantic annotation tasks such as entity linking (EL) [29, 30] and word sense disambiguation (EL) [31] could be applied to extract such features. Figure 2 shows three text documents taken from the BBC News Website,[7] with some of their words (in bold) disambiguated to specific concepts or named entities (represented by rectangles), which can be described in a knowledge graph (KG) [2, 32], i.e., a multi-relational graph that stores facts about concepts, entities and their semantic relations. These facts are represented as triples of the form $\langle node, link, node \rangle$, in which each node refers to a concept (class), named entity (instance) or literal, and each link represents a particular semantic relationship between nodes. The semantic annotations (linking words to KG nodes) in Fig. 2 were created by submitting the texts to the annotation tool Babelfy,[8] which linked words of the texts to nodes of the KG Babelnet.[9]

The word stem "Apple" appears in the three documents, but refers to a distinct thing in each one. Its disambiguation can be done according to the respective text context. In the text at the top left corner of Fig. 2, the word "Apples" refers to the *fruit*, which is said to be rich in some chemicals that can be found in certain kinds of food. Therefore, it is disambiguated to the word represented by the grey box, which is described in WordNet.[10] Considering this link and the remaining of the text, it is possible to classify this text in the category *Health*. On the other hand, the word "Apple" in the text at the top right was linked to the entity *Apple inc.*, which is a technology company, as the text also mentions the product *iPhone*, produced by *Apple inc.*, as represented by the link between them. It is possible to use these semantic annotations to classify this text in the category *Business*. Lastly, the text at the bottom was classified as *Entertainment & Arts* because several of its words were linked to entities related to music, namely the streaming service *TIDAL*, the record label *Apple Records* and the rock

[7] https://www.bbc.com/news

[8] http://babelfy.org/.

[9] https://babelnet.org/.

[10] http://wordnetweb.princeton.edu/perl/webwn?s=apple.

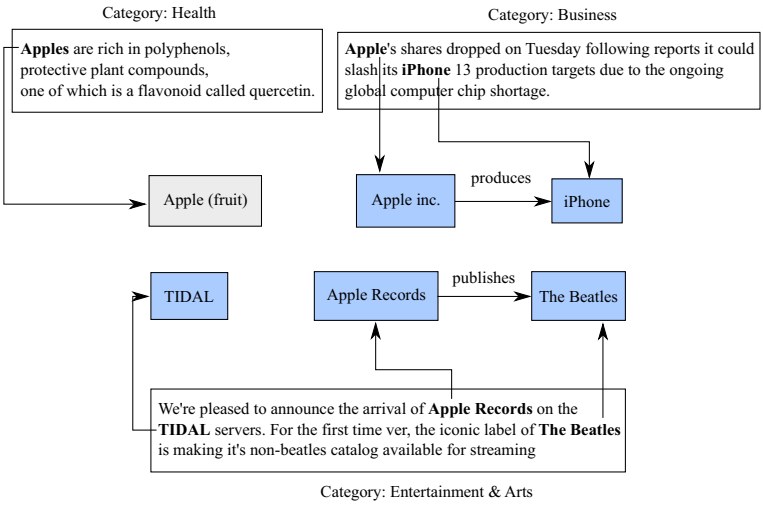


Fig. 2 Three examples of BBC text documents, their respective categories and some semantic annotations of their contents based on a KG. The grey box represents a concept and the blue boxes represent named entities

band *The Beatles*. The categories used to classify these texts are those of the BBC News Website.

This example illustrates how precise semantics can be essential for text classification in some challenging situations, like classifying short texts in a wide range of categories. Correctly capturing subtle semantic distinctions from texts with little contextual information can be key for proper classification. There are several ways to capture and represent semantics as features, and a variety of algorithms and techniques that can use these features in automatic approaches for text classification are discussed in the following.

3.2 Key aspects of text classification approaches

Some key aspects for the comparative analysis of text classification approaches have been devised from our bibliographical review, and concrete examples of text classification like the one just presented. These aspects include the characteristics of the text classification problem at hand, the classification methods employed, what data inputs and data features are considered, and feature representation. These aspects are used in this survey to help distinguish, classify, and compare text classification approaches from the literature, besides providing insights about promising research directions. They can be described in more detail as follows.

Classification problem characteristics While binary classification is the most common for texts, current industry needs are putting pressure to go beyond. Binary classification can already be challenging by itself, depending on the domain. However, text classification may happen in scenarios that involve multiple categories. For large sets of categories, a hierarchical structure is usually present. Taking advantage of such a structure during the learning and prediction processes defines what hierarchical classification is about. However, there are still some challenging issues, such as multi-class classification with many specific categories having subtle distinctive traits, unbalanced class distributions

in the training datasets, and limitations in the labeled data available for training the classification model.

Classification techniques One of the most important issues in text classification is choosing the classification technique. Without a conceptual understanding of each technique, one may not effectively determine the most suitable one for a given application. Some of the most used and recommended machine learning techniques for text classification are naive Bayes, decision tree, support vector machine (SVM), and neural networks. Nowadays, it is well known that neural networks and particularly deep learning approaches have achieved surpassing results on tasks such as image classification, face recognition, and several NLP tasks, including text classification. Nevertheless, some applications can be challenging, and the machine learning and deep learning areas are still evolving, with new alternative architectures arising frequently.

Data inputs and data features Different text classification approaches can have distinct inputs. Some approaches are adapted to classify specific kinds of text documents (e.g., microblog posts, news articles, academic papers). Thus, there are different domains with their peculiarities. For example, social media posts usually have short texts and present a more informal language with plenty of noise (typos, grammar errors, slang, acronyms, hashtags, etc.). Meanwhile, news articles and academic papers usually have longer documents with more formal text. In some cases, especially when context information is limited, as frequently happens with microblog posts, it is hard to grasp precise semantics from the texts.

Feature representation Machine learning and deep learning methods, including those usually employed for text classification, require input features represented as fixed-length vectors of numbers. Techniques to extract features from text and represent them as vectors range from bare counting of word occurrences in a text document to techniques that measure word relevance, capture semantics and reduce the dimensionality of the vector representations while keeping important properties. In this survey, we focus on the use of embeddings for feature representation. In the following, we describe vector representations in general and the major kinds of embeddings that can be useful in text classification.

3.3 Embeddings as features for text classification

Since texts have no explicit features, much work has aimed to develop effective text representations [33]. Some feature representations are more sophisticated and carry more information than others, what impacts the performance of a NLP model [34]. In addition, features extracted from text must be converted into a format that a machine can manipulate efficiently [35]. Thus, representation plays a vital role in many NLP-based tasks, such as text classification and clustering.

In conventional text classification, a document is usually represented based on the bag-of-words (BoW) model [36]. This simplistic model represents each text document d as a vector v, whose dimensionality is the number of words of the vocabulary V and each numeric value $v[i](1 \leq i \leq |V|)$ quantifies the importance of a word $w_i \in V$ to the document d (e.g., word w_i frequency in the document d (TF), inverse document frequency for w_i in d (IDF), TF*IDF) [37, 38]. Consequently, the BoW model frequently leads to sparsity (due to words that do not appear in certain documents) and scalability problems (due to the size of the vocabulary V, which can include thousands of words). Furthermore, the BoW model ignores word order, word senses (which can be determined by neighboring words in the fine context

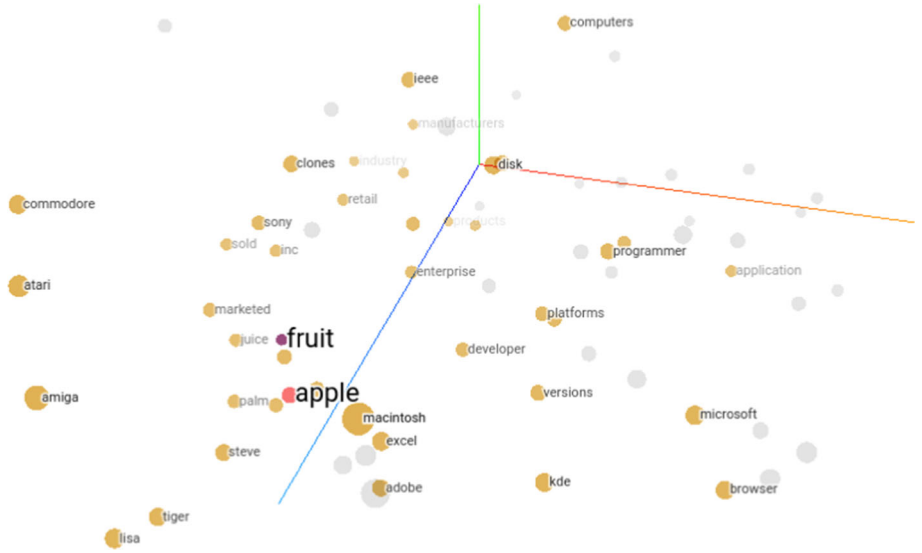


Fig. 3 Three-dimensional projection of Word2Vec embeddings of some words

where each word occurs), and semantic relations between senses (which can be represented in a knowledge graph, for example).

More recently, embeddings emerged as a means to circumvent these problems and improve the performance of text classification, among several other tasks. Embeddings encode complex data, such as words, word senses, or KG nodes and their relations in compact vectors (usually with a few hundred dimensions) that are suitable for efficient processing [39]. Embeddings aim to represent features in continuous vector spaces, preferably low-dimensional ones [12, 40]. Consequently, embeddings help to improve the results of text classification (among other NLP tasks) while maintaining the scalability of classification approaches for large amounts of texts [41, 42]. Ideally, an embedding should represent the data without loss of information, i.e., embeddings aim to preserve relevant or useful properties of the original data. In theory, a good embedding would produce exactly the same data when converted back to their original representation [43]. However, in practice, losses and distortions of the original data can occur. Thus, the goal is to preserve in the embedded representation what is relevant for some purpose (e.g., structural and/or semantic properties).

Usually, features that are alike keep, among other properties, a short distance between them in the embedded space. Figure 3 illustrates this in an example with word embeddings produced by the Word2Vec technique [13], which is based on a neural network model. The Embedding Projector[11] was used to further reduce the dimensionality of the 200-dimensional Word2Vec embeddings and generate the 3-dimensional visualization presented in Fig. 3. The embeddings of words with similar senses are close to each other. For instance, the embedding of the word *apple*, which can refer to the fruit or a corporation, as discussed in the text classification examples of Sect. 3.1, is close to those of both words *fruit* and *macintosh*.

Figure 4 presents a taxonomy of the major kinds of embeddings that have been or can be used to represent features potentially useful for text classification. *Text embeddings* are usually produced by applying non-supervised learning techniques to huge volumes of texts

[11] https://projector.tensorflow.org/.

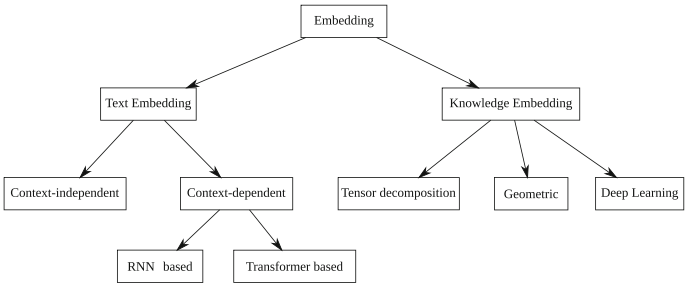


Fig. 4 Taxonomy of embeddings that can be useful for text classification

from sources like Wikipedia and news. Text embeddings can be *Context-independent* or *Context-dependent*. The former have a unique vector representation for each lexicon, while the latter can have distinct representations for each word occurrence, to capture sense variations in accordance with the textual context. Consequently, context-dependent embeddings can distinguish word senses such as *apple (fruit)*, *Apple Inc.* and *Apple Records*. Recently proposed context-dependent embeddings have yielded considerable performance gains over context-independent embeddings in several NLP tasks, including text classification. *Knowledge Embeddings*, on the other hand, codify knowledge such as nodes and relationships of a KG in representations based on vectors and/or tensors. They can be further classified according to the techniques used to generate them, such as *Tensor decomposition*, *Geometric*, and *Deep Learning*.

Currently, the most competitive approaches for text classification employ context-dependent embeddings of text, especially word embeddings. Their current popularity in text classification approaches is highlighted in Sect. 5. However, despite their wide popularity, text embeddings, even context-dependent ones, do not precisely represent semantic relations between concepts and entities as KGs do. In fact, the scientific community in the area of NLP is currently discussing what exactly textual embeddings capture from the texts used for training them, and if additional knowledge from external sources is necessary to properly solve some NLP tasks, as human beings do in some situations [44]. Thus, knowledge embeddings can theoretically become another alternative to represent useful features to classify texts with little contextual information, for example. These embeddings provide a suitable compact representation of facts present in a KG, about things that can be just mentioned in a text. Nevertheless, the potential of KGs and knowledge embeddings for leveraging text classification remains an open research question.

3.4 Effective combinations of classification techniques with embeddings

The vast majority of the text classification approaches found in the literature employ supervised learning, including generative (e.g., naive Bayes) and discriminative (e.g., SVM, random forest, neural networks) techniques. Figure 5 shows the most effective combinations of classification techniques (white rectangles) and embeddings (blue rounded rectangles) that we have identified in our review of the text classification literature.

The classification techniques in the leaves of the tree presented in Fig. 5 achieved the best performance and accuracy above 90% by using the embeddings indicated below the respective leaves as features to classify texts from at least one dataset listed in Table 5. These best results were obtained by using naive Bayes with bag-of-embeddings, random forest with

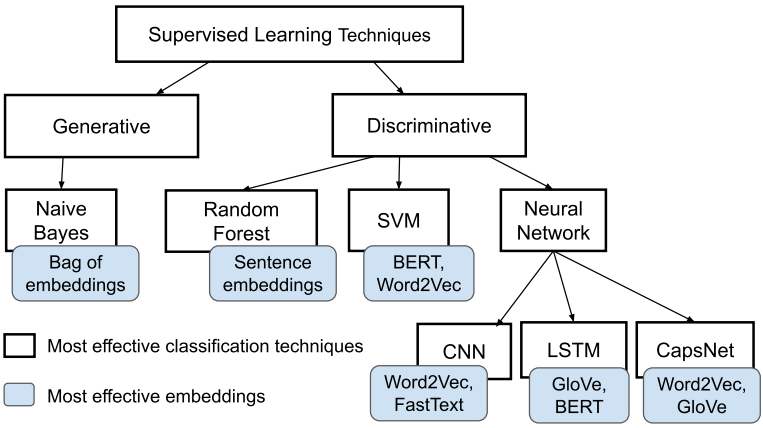


Fig. 5 Most effective combinations of classification techniques and embeddings identified in the text classification literature

sentence embeddings, SVM with BERT and Word2Vec; CNN with Word2Vec and FastText, LSTM with BERT and GloVe, and Caps net with Word2Vec and GloVe.

Section 4 describes and discusses some details of the text classification approaches selected from the literature and analyzed to draw these conclusions. Then, in Sect. 5.1, Table 1 lists the classification techniques and embeddings employed by each approach, besides the domain of the texts they are intended to, while Tables 5 and 6 list the performance measures of these approaches on the datasets they were evaluated with, according to the results available in the papers that introduced the respective approaches.

4 Text classification approaches using embeddings

This section reviews text classification approaches that explicitly use embeddings of any kind as features for text classification. The reviewed works were collected in an extensive and systematic review, as explained in Sect. 2. They are described and analyzed in the following subsections, according to the kinds of embeddings employed and their chronological evolution. Their classification techniques, premises, purposes and achieved performance are presented and discussed, and summarized conclusions are provided in the end of each subsection. The first 3 subsections refer to the embedding models that are most prominent in the text classification approaches that we have analyzed, namely Word2Vec, Glove and BERT. The following subsection examines works that use less popular embedding models. Works using more than one embedding model are described in the subsection referring to the model that we consider paramount in the proposed solution, if we identify such a model. Otherwise, that work is discussed in a final subsection about text classification approaches that use diverse embeddings or are produced by combining models.

Although most text classification proposals providing high performance are intended to binary classification, a considerable number of recent works from the literature tackle multi-class and multi-label classification. Thus, we consider in our review selected proposals for any of these classification problem variations. However, to keep the scope manageable, this work does not address particular classification problems and techniques, such as multi-level text categorization [45–47] and micro-word-based approaches [48], among other promising ones that have been investigated in the literature.

4.1 Approaches using Word2Vec

Word2Vec [13] is a statistical approach for learning word embeddings from a text corpus, developed by Tomal Mikolov with the intent to make neural network-based training more efficient. It has become a benchmark for developing pre-trained context-independent word embeddings, and is one of the most used for text classification approaches.

Lai et al. [49] present a recurrent convolutional neural network for text classification. Their model captures contextual information and builds feature representations using the recurrent structure, which may introduce less noise than traditional window-based neural networks. In addition, it uses a max-pooling layer that automatically judges which words play critical roles in text classification. The pre-training of word embeddings fed to the neural network uses the Skip-gram model. The experimental results show that the neural network approaches outperform the traditional methods for all four tested datasets. The highest accuracy rate achieved was 96.49% on the 20NewsGroup dataset. The authors affirm that it was possible due to the combination of the recurrent structure and the max-pooling layer, and the advantages of both neural models, recurrent and convolutional. The first is effective in capturing contextual information, and the latter may better capture the semantics of texts compared to recursive or recurrent neural networks.

Lenc et al. [50] investigated the use of word embeddings for representing long texts in multi-label classification. The embeddings were used in three convolutional neural network topologies. The authors analyzed the semantic similarity of the embedding vectors learned during the network training and compared them with the standard word2vec vectors. The precision score achieved was 91.03%. It was observed similar results for both variants learned by CNN. They concluded that initialization of the embeddings with Word2Vec pre-trained vectors does not play an essential role for text classification. However, learning word vectors is crucial to obtain good results.

Zhao et al. [51] exploited capsule networks with dynamic routing, a variant of the capsule networks introduced in [52]. They proposed three strategies to stabilize the dynamic routing process to alleviate the disturbance of some noise capsules, which may contain background information or have not been successfully trained. The basic idea of dynamic routing is to construct a nonlinear map in an iterative manner, ensuring that the output of each capsule is sent to an appropriate parent in the subsequent layer. They also show that capsule networks significantly improve classification over competitors specially for doing multi-label instead of single-label classification. Their model substantially and consistently outperforms simpler deep neural networks such as LSTM, Bi-LSTM, and CNN-rand by a noticeable margin on all the experimental datasets. Their capsule network model also achieves competitive results against more sophisticated deep learning models such as LR-LSTM, Tree-LSTM, VC-CNN, and CL-CNN. The approach achieved 93.8% of the accuracy score. However, though the capsule network exhibits high performance in single-label text classification, multi-label text classification is a more challenging problem because more training is required to cover the whole label space. Some works [49, 53–55] that employ Word2Vec for text classification focus on semantic characteristics and contextual information.

Guo et al. [56] proposed a novel term weighting scheme to be combined with word embeddings to enhance the classification performance of CNNs, considering the fact that one term generally has different importance in documents with distinct class labels. In this scheme, multiple weights are assigned to each term, and these weights are applied separately to the embedding of each word. The weighted word embeddings generated are fed to a multi-channel CNN model with the above-obtained weight vectors to execute classification. The input of each channel is obtained by applying the weight vectors corresponding to

each class to the word embedding matrix, while in the original Kim's CNN model [57], the inputs for two channels are two word embedding matrices obtained by CNN-static and CNN-non-static schemes, respectively. Their experiments with publicly available Word2Vec vectors calculated the weights with training data only and trained the multi-channel TextCNN model using mini-batches, with each batch consisting of 50 documents. By comparing the novel method with several other baselines methods with five benchmark data sets, the results manifest that the classification accuracy of the proposed method exceeds that of other methods by a considerable margin. The highest accuracy rate achieved was 95.6% on the Subjectivity dataset.

Liu et al. [53] proposed the Task-oriented Word Embedding method (ToWE), which was specially designed to capture semantic and task-specific features of words for text classification. It introduces the function-aware component, which highlights the word's functional attributes in the embedding space by regularizing the distribution of words to have a precise classification boundary. This proposal uses Word2Vec to model the context information and log-linear models to produce word embeddings. The authors conclude that their method performed better than compared methods. In particular, the ToWE-SG version of the method significantly outperforms the other baselines on the tested datasets, achieving 90.8% of the accuracy score in IMDB dataset.

Differently from [53], Pan et al. [54] proposed the Simple Word Embedding-based Model (SWEM) for text classification, which uses a modified hierarchical pooling strategy for simple word embedding in the few-shot transfer learning style. The model leverages and transfers knowledge obtained from some source domains to recognize and classify unseen text sequences with just a handful of support examples in the target problem domain. Extensive experiments using SWEM with Word2Vec and SVM to classify texts in English and Chinese from five datasets, reaching 92.1% of the accuracy score in DBpedia dataset, demonstrated that SWEM with parameter-free pooling operations is able to abstract and represent textual semantics useful to improve the results of text classification.

Meanwhile, Shi et al. [58] consider the fusion of artificial neural network models to overcome the limitations of using only textCNN or LSTM for classifying news. They proposed a C-LSTM with word embedding model to obtain more accurate features considering the context information. This model extracts local features from the document through textCNN and global specialties referring to the whole document through LSTM. To preserve the original document better, the model integrates the word embedding into the fusion layer. As the most classical model, textCNN acquired a high classification accuracy and fast speed. However, the proposed method C-LSTM achieved the best result in experiments applied to the Chinese news dataset, achieving an accuracy rate of 86.53%.

Pittarras et al. [55] applied semantic augmentation to enhance semantics of the inputs of deep neural networks used for text classification. Their proposal selects frequency-based semantic information from the WordNet semantic graph and fuses it with deep neural embeddings. For each word, it extracts semantic information from an appropriate source of existing knowledge, such as a semantic graph. It generates a vector for each word and represents the whole text as a fusion of the word semantic vectors. This approach achieves the best average performance when using raw concept frequencies, selecting the first retrieved concept per word (basic strategy), and concatenating the resulting vector to the Word2Vec embedding. The highest accuracy value reached was 97.6% in BBC corpus. They concluded that the way of introducing semantic information to the model affected training and the performance of the learned model.

Liu et al. [59] presented a model for multi-label text classification with many classes, called joint learning from Label Embedding and Label Correlation (LELC). It is based on

the multi-layer attention and label correlation. LELC considers both the co-occurring label matrix and the label correlation matrix to exploit the potential label information and label correlation. Firstly, it uses a bidirectional gated recurrent unit network (BIGRU) to extract basic features, capture text contents information and sequence information at the same time, and then apply the multi-layer attention framework to facilitate selecting valid features related to labels. Then, the label correlation matrix is taken into account in the process of performing label space dimension reduction (LSDR), which is fundamental for multi-label learning to simplify the process of model learning. At last, deep canonical correlation analysis technology was used to couple features and latent space in an end-to-end pattern. Experimental results in real-world datasets demonstrate the effectiveness of the model, achieving 92.22% of the Macro F1 score.

Finally, Gallo et al. [60] presented a method that map text features to an image vectors to take advantage of image neural models to classify text documents. They convert each text document into an encoded image by using word embeddings. Analogous to word embedding models that can produce similar word embeddings for words occurring in equivalent contexts, their approach exploits this property to transform a text document into a sequence of colors (visual embedding), obtaining an encoded image that keeps similarity with the image encoding of similar documents. Their approach computes the Word2Vec word embedding of a text document, quantizes the embedding, and arranges it into a 2D visual representation as an RGB image. As a result, the method achieved competitive performance on well-known benchmark text classification datasets. The evaluated metric was percentage error, achieving 1.07% in DBpedia dataset.

Notice that, most of the text classification approaches previously discussed are based on deep neural networks. The capsule network [52] has exhibited a better-transferring capability than conventional deep neural networks, namely recurrent neural networks (RNNs) and convolutional neural networks (CNNs). On the other hand, just a few works using Word2Vec specifically addressed multi-label text classification [50, 59].

4.2 Approaches using GloVe

Global Vectors for word representation (GloVe) [14] uses the matrix factorization technique to embed the word-context matrix. This large and usually sparse matrix maintains for each word w of the vocabulary V a measure of the co-occurrences of each other word of $V \backslash \{w\}$ in the textual context (close neighborhood consisting of a few words on each side of some occurrence of w) in some corpus. The idea behind this matrix is to derive the relationship between words from statistics of their co-occurrences in local text contexts. The approach is similar to the Word2Vec method, where each word is presented by a high dimension vector and trained based on the surrounding words over a huge corpus.

GloVe has been successfully used in text classification approaches [61–64]. Chalkidis et al. [62] released a new dataset of 57k legislative documents from EUR-LEX, annotated with approximately 4.3k EUROVOC labels, intended to zero-shot learning, but applied in large multi-label text classification. They reported several experiments with different neural classifiers using pre-trained GloVe embeddings. They found out that a BIGRU with label-wise attention performs better than some state-of-the-art methods. They also investigated which zones of the documents are more informative on the dataset EURLEX57K, showing that considering only the title and recitals of each document leads to almost the same performance as viewing the entire document. The approach achieved 69.8% of the Macro F1 score. One

major limitation of the investigated methods is that they are unsuitable for extreme multi-label text classification, with hundreds of thousands of labels.

Zhang et al. [61] proposed a novel CNN-based two-phase framework together with data augmentation and feature augmentation for recognizing text documents of classes that have never been seen in the learning stage (the so-called zero-shot text classification). It applies GloVe vectors as word embeddings. In fact, four kinds of semantically rich features (word embeddings, class descriptions, class hierarchy, and a general knowledge graph) are incorporated into the proposed framework to deal with instances of unseen classes effectively. The approach achieved 85.2% of the accuracy score. Therefore, the experiments show that data augmentation by topic translation improved the accuracy in detecting instances from unseen classes. In contrast, feature augmentation enabled knowledge transfer from seen to unseen classes for zero-shot learning.

Kim et al. [63] presented an application of capsule networks to the text classification domain and suggested utilizing a static routing variant to reduce the computational complexity of dynamic routing. Capsules consider the spatial relationships between entities and learn these relationships via dynamic routing. For this characteristic, capsules applied to text classification had advantages over the tested convolutional neural networks. The experimental results indicated that the achieved accuracy of 94.8% from the static-routing model is higher than that of the dynamic-routing model. Furthermore, the proposed model results were comparable results to those of initial studies regarding capsule network-based text classification.

Moreo et al. [64] proposed word-class embeddings (WCEs), i.e., distributed representations of words specifically designed for multi-class text classification. They showed that, when concatenated to pre-trained word embeddings, WCEs substantially facilitate the training of deep-learning models for multi-class classification by topic. They also showed empirical evidence that WCEs consistently improve multi-class classification accuracy, using six popular neural architectures and six widely used and publicly available datasets for multi-class text classification. Their method does not involve any optimization procedure but operates directly on the co-occurrence counters. The approach achieved 73.1% of the F1 score in Ohsumed dataset.

According to the analyzed works, it is possible to observe different classification types as multi-class and multi-label text classification. In addition, distinct metrics (Macro F1, F1, Accuracy) were used to evaluate method performances in varied classification techniques. Therefore, the capsule network [52] achieved better performance in terms of accuracy.

4.3 Approaches using BERT

Bidirectional encoder representations from transformers (BERT) [16], based on the transformer architecture, is a big neural network architecture with a huge number of parameters, that can range from 100 million to over 300 million. Several recent works have further studied and improved the BERT objectives and architecture and used it for different purposes. Some works applied it to text classification [65–68].

Cai et al. [65] proposed a hybrid BERT model that incorporates Label semantics via Adjustive attention (HBLA). It is a hybrid neural network model to take advantage of both label semantics and fine-grained text information to improve classification. The approach creates the label correlations graph based on adjacency similarity and encodes this graph to capture structure information and correlations among labels. The proposal also includes the design of a novel attention mechanism called adjustive attention to measure the semantic relation between word and label. BERT was used to capture the label-related discrimination

information from each document and to obtain the implicit representation in each word's context. The experimental results achieved 90.6% of the precision score.

Meng et al. [66] presented the Label-Name-Only Text Classification (LOTClass) model to explore the potential of using only the name of each class to train classification models on unlabeled data, i.e., without using labeled documents. Pre-trained neural language models are used as general linguistic knowledge sources for category understanding and as representation learning models for document classification. This method associates semantically related words with the label names, finds category-indicative words, trains the model to predict their basic categories, and generalizes the model via self-training. The effectiveness of LOTClass is assessed on four benchmarks. However, there are complex cases where label names are insufficient to build the correct classification model. The approach achieved 91.6% of the accuracy score in Amazon Review dataset.

The Out-of-manifold Regularization in Contextual Embedding Space for Text Classification (OoMMix), proposed by Lee et al. [67], is a new approach to find and regularize the remainder of the space to address the over-parameterization problem, referred to as out-of-manifold. The motivation is that the embeddings computed from the words only utilize a low-dimensional manifold, while a high-dimensional space is available for the model capacity. Therefore, OoMMix discovers the embeddings that are useful for the target task but cannot be accessed through the words. Precisely, they synthesize the out-of-manifold embeddings based on two embeddings obtained from actually observed words to utilize them for fine-tuning the network. A discriminator is trained to detect whether an input embedding is located inside the manifold or not, and simultaneously, a generator is optimized to produce new embeddings that can be easily identified as out-of-manifold by the discriminator. In the end, the fine-tuning on the synthesized out-of-manifold embeddings tightly regularizes the contextual embedding space of BERT. The experimental results achieved the accuracy score of 99.03% in DBpedia dataset.

Jiang et al. [68] proposed the Light deep learning model (LightXML) a fine-tuning of the single transformer model with dynamic negative label sampling. To make LightXML robust in predicting, they proposed dynamic negative sampling based on these generative cooperative networks to recall and rank labels. With generative cooperative networks, the transformer model can be end-to-end fine-tuned in extreme multi-label classification, which makes the transformer model learn powerful text representation. The dynamic negative sampling allows label ranking part to learn from easy to hard and avoid overfitting, which can boost overall model performance. Experiments showed that LightXML outperforms state-of-the-art methods in five extreme multi-label datasets. Although their experiments showed good results, 96.77% of the accuracy score, their model has a much smaller size and lower computational complexity than current state-of-the-art methods.

The presented approaches achieved satisfactory results. Some of them applied neural networks as a classification technique. However, regardless of the method used, all approaches obtained accuracy above 90%. In conclusion, the use of BERT enables preeminent model performance over legacy methods and an ability to process larger amounts of text and language.

4.4 Approaches using other embeddings

Besides Wor2Vec, GloVe and BERT, other text embeddings have been used in a few text classification approaches. These embedding models include FastText [15], Doc2Vec [69], and region embedding [70], among others. Sparse Local Embeddings for Extreme Classification

(SLEEC) was proposed by Bhatia et al. [71] as an extreme multi-label classifier to address the problem of learning a classifier that can automatically tag a data point with the most relevant subset of labels from a large label set. The SLEEC algorithm extends embedding methods. Its main technical contribution is a formulation for learning a small ensemble of local distance preserving embeddings that can accurately predict infrequently occurring labels. During prediction, SLEEC uses a k-nearest neighbor (kNN) classifier in the embedding space, thus leveraging on the preservation of nearest neighbors during training. The experimental results achieved 85.54% of the precision score in Wiki10 dataset.

Qiao et al. [70] presented a new learning method and utilized distributed representations of n-grams, referred to as "region embeddings", for text classification. The regions in a document, in this case, were considered as fixed-length contiguous subsequences of tokens in the document. The approach focused on learning the representations of small text regions, which preserve the local internal structural information for text classification. For utilizing the word-specific influences of each word on its context words, a local context unit for each word is learned in addition to word embedding. They used the interactions between words and their local context based on word embeddings as well as the local context units to produce region embeddings. In addition to the analysis of region sizes, they further studied the influence of word embedding dimensions. The experimental results achieved 98.9% of the accuracy score in DBpedia dataset.

The works [32] and [72] use FastText embeddings in their text classification models. Hossain et al. [72] introduced a convolution neural network-based model using FastText embedding for text document classification of resource-constrained languages. A corpus of documents in a low-resource language, namely Bengali, was developed to assess the performance of the proposed model, and different hyperparameters of the CNN model were tuned for optimization and hence to achieve better classification results. Despite the challenge to develop an automatic text classification system for low-resource languages such as Bengali, with a scarcity of digital resources and benchmark corpora, evaluation results on test datasets showed improved performance of the proposed method compared to existing techniques, such as TF-IDF-SVM, Word2Vec-SVM, GloVe-SVM, and FastText-SVM.The highest accuracy score achieved was 96.85%.

Aly, Remus and Biemann [32] applied simple shallow capsule networks for hierarchical multi-label text classification and showed that they can perform superior to other neural networks, such as CNN's and LSTMs, and non-neural network architectures such as SVMs. Results of experiments with pre-trained FastText embeddings adjusted during training confirmed the hypothesis that capsule networks are especially advantageous for rare events and structurally diverse categories. The approach achieved 82.75% of the precision score. Therefore, the main benefit of capsules shown is their ability to encode information of each category separately, by associating each capsule with one category. Combining encoded features independently for each capsule enables capsule networks to handle label combinations better than previous approaches.

Pappas and Henderson [73] proposed a novel joint input-label embedding model for neural text classification that generalizes over existing input-label models and addressed their limitations while preserving high performance on seen and unseen labels. Models were evaluated on full-resource and low or zero resource text classification of multilingual news and biomedical text with a large label set. Two nonlinear transformations address the need to capture complex label relationships with the same target joint space dimensionality. The experimental results achieved 79.85% in the F1 score. Therefore, the approach differs from others due to the ability of the model to capture complex input label relationships, controllable capacity, and training objective, which is based on cross-entropy loss.

Also based on region embedding, Li and Ye [74] presented a text classification model combining region embedding and a RELSTM to form a hybrid neural network. The RELSTM first divides regions for text and then generates region embeddings. The input of the RELSTM model is a word index, where each word embedding is initialized and continuously adjusted during the training process. This model does not require pre-trained word embeddings, what simplifies the experimental process and facilitates the investigation of the impact of region size, the number of layers and hidden nodes on the accuracy of the RELSTM. The highest accuracy score achieved was 97.7% in Paper Theme dataset

Finally, Chang et al. [75] proposed X-Transformer, a scalable approach to fine-tuning deep transformer models for text classification, using a pre-trained XLNet [76]. X-Transformer consists of a Semantic Label Indexing component, a Deep Neural Matching component, and an Ensemble Ranking component. The proposed method achieves new state-of-the-art results on four benchmark datasets. The highest precision score achieved was 96.7% in AmazonCat-13k. However, successfully applying transformer models to extreme multi-label problems remains an open challenge due to the vast output space and severe label sparsity issues.

4.5 Approaches using embedding combinations

Several approaches combine distinct models to create embeddings for text classification. Xu et al. [77] proposed the Topic-based Skip-gram approach to learn topic-based word embeddings and two CNN architectures that use multiple word representations simultaneously for text classification. This approach uses latent Dirichlet allocation (LDA) to capture precise topic-based word relationships and integrate them into distributed word embedding learning with a novel objective function. The approach results achieved 95% in the Macro F1 score in 20NewsGroup dataset.

Jin et al. [78] built a text classifier using a naive Bayes model with a bag-of-embeddings that extends the skip-gram model [13] to incorporate context and word sense information. To better do this, they train multi-prototype target word embeddings, with one distinct vector trained for a word under each class, i.e., each word embedding is produced separately. The context vectors of words remain the same across different classes. Compared with bag-of-word models, the bag-of-embeddings model exploits contextual information by deriving the probabilities of class-sensitive embedding vectors from their inner product with context words. The proposed model achieved 96.5% of the accuracy rate.

Most of the existing methods for multi-label classification learn a single linear parametrization using the entire training set. Hence, they fail to capture nonlinear intrinsic information in feature and label spaces, due to the exponential size of the output space. To overcome this, Kumar et al. [79] presented a new multi-label classification method, called MLC-HMF, which learns piecewise-linear embeddings with a low-rank constraint on parametrization to capture nonlinear intrinsic relationships that exist in the original feature and label space. The approach considered accuracy as the evaluated metric, achieving 93%. Therefore, the experimental analysis provided evidence that hierarchical embedding can yield more accurate results for multi-label classification.

Wang et al. [80] investigated label embeddings for text representations and proposed the Label Embedding Attentive Model (LEAM) to improve text classification. LEAM was implemented by jointly embedding words and labels in the same latent space, and the text representations were constructed directly using text-label compatibility. They introduced an attention framework that measures the compatibility between embeddings of text sequences and labels. The attention is learned on a training set of labeled samples to ensure that, given a

text sequence, the relevant words are weighted higher than the irrelevant ones. Their method maintains the interpretability of word embeddings and enjoys a built-in ability to leverage alternative sources of information, in addition to input text sequences. Experiments were done in 6 different datasets, achieving 99.02% of accuracy in DBpedia dataset

Liu et al. [81] presented a distributional representation of words combined with their part-of-speech tags. This embedding model is a modification of the continuous bag-of-words model that predicts the current word based on the context [13]. They build a two-dimensional look-up table on the training set in the format $\langle word, part\_of\_speech \rangle$ to represent word features. It makes word embeddings more expressive and semantically rich, improving performance of a Bayesian classifier. If a new category is added, the model does not need to recalculate the word embeddings for this category. The experimental results achieved 90.2% of the accuracy score in DBpedia dataset.

Sinoara et al. [82] proposed two models to represent document collections based on both words and word senses, having the objective of improving text classification performance through enriching text representations with semantics. They use word sense disambiguation tools and available pre-trained word and word sense models to construct embedded representations of documents. The proposed approach has the potential to be applied to documents written in several languages, since it relies on the multilingual Babelnet KG and pre-trained word embeddings. Their representations are low-dimensional, what helps to speed up the learning and the classification processes. Their experimental evaluation indicates that the use of the proposed representations provides stable classifiers with reliable quantitative results, especially in semantically complex classification scenarios. The highest Macro F1 score was 97.29% in BBC Corpus.

Le and Mikolov [13] introduced Word2Vec and also the doc2vec method for learning embeddings of phrases. They inspired embeddings for coarser textual granularities [69] and their application to text classification. For example, [83] propose a rule-based approach using doc2vec embeddings in text classification. Their work investigates how varying rule-based classification and embeddings of distinct text granularities influence performance. Their document vector rule-based (D2vecRule) proposal, tested on the datasets Reuters-21578 and 20 Newsgroups, achieved 90.72% of accuracy in the first dataset, and good results according to the F-measures and implementation time metrics.

Gupta et al. [84] proposed a novel document representation technique, Sparse Composite Document Vector Multi-Sense (SCDV-MS), extended from SCDV to consider the multi-sense nature of words. SCDV-MS utilizes multi-sense word embeddings and learns a lower-dimensional manifold. Experiments on multiple real-world datasets showed that SCDV-MS embeddings outperform previous state-of-the-art embeddings on multi-class and multi-label text categorization tasks. The accuracy score achieved was 86.19%. Furthermore, comparing the results, SCDV-MS embeddings proved efficient in time and space complexity for textual classification.

Bounabi et al. [85] presented a study that allows understanding the advantages of Doc2vec and profit from them in neural embedding applications. They use Doc2vec to produce vector inputs for machine learning models through the variant Paragraph Vector-Distributed Memory (PV-DM). Then, it is incorporated into hybrid ML methods to improve the classification quality. Different parameters control the effectiveness of the document representation. Experimental results prove that the architecture based on the PV-DM version with the average method, plus an optimal epoch number and minimal vector size, has a positive impact on text classification, achieving 99.1% of accuracy .

Hu et al. [86] proposed an enhanced word embedding method that introduces a unique sentence reorganization technology to rewrite all the sentences in the original training corpus.

Then, the original and the reorganized corpora are merged in a training corpus of the distributed word embedding model to solve the coexistence problem of words and phrases in the same vector space. The advantage of this method is that it can incorporate phrase features into the original vector space to form a hybrid distributed embedded structure where words and phrases coexist. In addition, it does not need any additional training corpus. Consequently, it can also alleviate the problem of insufficient word embedding corpus for training. The experimental results achieved 97.23% of the accuracy score in R52 dataset.

Liu et al. [87] proposed a Label-Embedding Bi-directional Attentive model to improve the performance of the BERT text classification framework, extending this framework with label embedding and bi-directional attention (Text-to-Label Attention and Label-to-Text Attention). Therefore, it was built upon BERT, whose outputs are the sequence-level text representation and the token-level text representation. The bidirectional attention mechanism was proposed to integrate information between label embedding, sequence-level text representation, and token-level text representation. Experimental results demonstrated the effectiveness of their proposal, achieving 94.4% of the Macro F1 score.

Zhang et al. [88] proposed an approach to incorporate knowledge about class labels into text classification models. In this approach, label-related knowledge is represented by keywords that users can customize. The relatedness between each word in the text sequence and hidden knowledge, such as keywords, is calculated and concatenated with the original model's information. Their proposal showed to be capable of understanding the relationship between sequences and labels, performing well on datasets with many classes.The highest accuracy score achieved was 94.72% in AG News dataset.

Finally, [89] proposed the Language Agnostic Sentence Representations (LASER), which uses a single encoder to generate an embedded vector per sentence in any language. This approach focuses on purpose classification in different speeches, given that the model is trained only in one language. Several machine learning models were developed, and their outputs were compared to understand how zero-shot learning (ZSL) works. The highest accuracy score achieved in the experiments was 93%. The approach is grounded on the idea that the sentence embedding of two sentences of different languages having the same sense must be similar. Therefore, the effectiveness of this model can be tested when two different terminologies are used in the same sentences, i.e., writing words of one language in the alphabets of a second language or when there are two different sets of alphabets belonging to two different languages in the same sentence.

5 Comparative analysis

Table 1 provides a comparison summary of the text classification approaches found in the literature and described in Sect. 4 that use embeddings as feature representations. They are listed in chronological ordering, presenting the columns according with relevant aspects of text classification described in Sect. 3.2. The column *Domain* refers to the source and nature of the classified text documents. The *Method* employed by each work is indicated in the third column. Lastly, the column *Feature Representation* denotes the kind of feature representation used to classify the texts. Other methods and tools used to generate features or preprocess the data are not shown in this table because they are outside the scope of this paper.

The column *Domain* of Table 1 allows one to perceive that most of the works are multi-domain, i.e., they are intended to work, trained and evaluated with text from different domains. Furthermore, many works focus only on news articles. By analyzing Table 1 with the work

Table 1 Classification approaches using embeddings

Work	Domain	Classification Technique	Feature Representation
Lai et al. [49]	News article, Academic paper, Sentiment Analysis	RCNN	Word embeddings, Word2Vec
Bhatia et al. [71]	Multi-domain	SLEEC	Word embeddings
Xu et al. [77]	News articles, Medical articles	CNN	Word embeddings
Jin et al. [78]	News articles	Naive Bayes	Word embeddings, Bag-of-embeddings
Lenc et al. [50]	News articles	CNN	Word embeddings, Word2Vec
Qiao et al. [70]	News article, Reviews, All-purpose	Word-Context region embedding, Context-Word region embedding	Word embeddings, Region embedding
Liu et al. [53]	News article, Reviews, Academic paper, Sentiment analysis	ToWE	Word embeddings, Word2Vec
Zhao et al. [51]	News article, Reviews, Sentiment analysis	CapsNet	Word embeddings, Word2Vec
Kumar et al. [79]	Sentiment analisys, Medical article, News article	MLC-HMF	Word embeddings
Wang et al. [80]	News article, All-purpose	LEAM	Label-Embeddings, Glove
Aly et al. [32]	Book blurbs, Academic articles	CapsNet	Word embeddings, FastText
Zhang et al. [61]	News articles, All-purpose	CNN	Word embeddings, GloVe
Chalkidis et al. [62]	Legislative text	CNN, GRU, HAN	Word embeddings, GloVe, BERT
Pappas et al. [73]	News articles, Biomedical question	HAN, MHAN	Word embeddings, Label-Embeddings
Pan et al. [54]	News article, All-purpose	SVM	Word embeddings/ Word2Vec
Liu et al. [81]	News articles	LIBSVM	Word embeddings
Guo et al. [56]	Reviews, All-purpose	CNN	Word embeddings/ Word2Vec
Shi et al. [58]	News articles	C-LSTM	Word embeddings/ Word2Vec
Sinoara et al. [82]	News article, Computer Science Technical Reports, Biomedical text, Sentiment Analysis	Naive Bayes (NB), Sequential Minimal Optimization (SMO), Inductive Modelbased on Bipartite Heterogeneous Net-works (IMBHN)	Word embeddings/ Knowledge embeddings/ Babel2Vec / Word2Vec

Table 1 continued

Work	Domain	Classification Technique	Feature Representation
Cai et al. [65]	News articles, Academic paper	Bi-LSTM	Label embeddings/ BERT
Bounabi et al. [85]	Sport news	Support Vector Machine, Logistic Function, Feedforward neural network (FNN), Hybrid ML model	Word embeddings/ Doc2Vec
Hu et al. [86]	Movie review, Academic paper	CNN	Word embeddings
Li et al. [74]	News article, Hotel review	LSTM	Region word embeddings
Aubaid et al. [83]	News articles	JRip (RIPPER), One Rule (OneR) ZeroR	Word embeddings/ Doc2Vec
Gupta et al. [84]	News articles	LinearSVM, Logistic regression, SCDV-MS	Word embeddings/ Doc2vec
Meng et al. [66]	News articles, Movie review, All-purpose	LOTClass	Word embeddings/ BERT
Chang et al. [75]	All-purpose	X-Transformer	Word embeddings/ XLNet
Kim et al. [63]	News article, Reviews,	CapsNet	Word embeddings/ GloVe
Pittaras et al. [55]	News article, Biomedical text	DNN	Word embeddings/ Word2Vec
Liu et al. [59]	News article, Medical article, Reviews, email text	LELC	Word embeddings/ Wor2Vec
Liu et al. [87]	News article, Academic paper, Sentiment analysis	BERT classifier	Label embeddings/ BERT
Hossain et al. [72]	News article	CNN	Word embeddings/ FastText
Saraswat et al. [89]	Consumer's queries	Random Forest, Support Vector Machine and Multilayer Perceptron model	Sentence embeddings
Gallo et al. [60]	News article, Sentiment analysis, All-purpose	CNN	Word embeddings/ Word2Vec
Lee et al. [67]	News article, Reviews, All-purpose	SVM, BERT	Contextual embeddings/ Bert
Zhang et al. [88]	News article, Movie review	Bi-LSTM	Word embeddings/ Knowledge embeddings/ GloVe, BERT
Jiang et al. [68]	Multi-domain	Generative cooperative networks	Label embeddings/ BERT, XLNet, RoBERTa
Moreo et al. [64]	News article, Biomedical text	SVM, BERT, TCNN, LEAM, FastText classifier, CNN, LSTM, ATTN	Word embeddings/ GloVe

details described in Sect. 4, it is possible to notice that most corpora have long and formal texts. Moreover, several ones refer to everyday affairs. Such datasets are probably used because embeddings are usually trained with texts that also refer to everyday affairs. Lastly, the use of long corpora may be an indicative that the current approaches require a significant amount of context to be able to correctly categorize the text documents. From the analyzed works, only [89] tackles short documents, to categorize consumer's queries. The classification of short documents, be it informal (like social media posts) or not, is a challenging research theme.

Regarding the column *Method*, notice that most works use neural networks, with varying architectures, as the classifier. This happens because neural networks, and particularly deep learning, have surpassed the performance of machine learning algorithms on several tasks, including text classification [26]. Deep learning can execute featuring engineering on its own and promote fast learning. Although it is not explicit in Table 1, most recent works employ deep neural networks, a hot topic in several research domains nowadays due to their superior performance and current availability of the necessary hardware. As the approaches employ a myriad of neural network architectures, which are still evolving fast, we prefer not to draw any conclusion about the most used and suitable architectures yet. However, it is possible to notice that several approaches employ neural networks architectures that consider, in some way, the order in which the words occur.

The chronological order of the proposals in Table 1 allows noting, in its last column, *Feature Representation*, that the embedding approaches are shifting from context-independent word representations, like Word2Vec, to approaches that use contextual embeddings, based, for example, on the transformer technology, as BERT. However, Word2Vec is still the most used embedding, followed by Glove. This is an indicative that these embeddings are mature enough to be used in several domains.

Table 2 provides links to the repositories of open-source code (when available) or to the GitHub profiles of the author(s) of the approaches considered in this survey, to facilitate access to what was produced by each work. Section 5.1 provides a performance comparison summary of these approaches.

5.1 Evaluation of text classification approaches

The works analyzed in this survey use several datasets and performance metrics to evaluate their text classification approaches. Tables 3 and 4 provide pointers to the datasets used. Table 3 presents links to their respective home pages, when they are available. When no such link is unavailable, Table 4 provides a reference to the paper or challenge in which the dataset appears.

The most used metric to evaluate the performance of the approaches considered in this paper is the accuracy score. It is usually characterized in terms of error and is traditionally decomposed into bias (systematic error) and variance (random error) components. The next most used evaluation metric is by far the F1. It tolerates uneven class distributions better than accuracy. Therefore, depending on the dataset used to evaluate the text classification approach, the accuracy may provide a less reliable measure of the performance than F1. Table 5 presents the accuracy score of works analyzed in this survey (listed in the first column of the table, by the chronological order of their publications) that use this metric to evaluate performance on distinct datasets (listed in alphabetic order in the second line of the table header). Table 6 summarizes the performance of the works analyzed in this survey that, instead of the standard accuracy score, use other metrics (listed in the first column of the table) to evaluate their approaches. The highest performance value (accuracy, F1, micro F1,

Table 2 Links to the considered approaches. When available, a link to the source code repository is provided in the second column. Otherwise, the Github profile of each author is provided in the third column, on the respective full name

Work	Source code repository	Github of authors
Lai et al. [49]	https://github.com/roomylee/rcnn-text-classification	
Bhatia et al. [71]	https://github.com/xiaohan2012/sleec_python	
Xu et al. [77]	https://github.com/HaotianMXu/Multimodal-CNNs	Haotian Xu
Jin et al. [78]		
Lenc et al. [50]		
Qiao et al. [70]	https://github.com/schelotto/Region_Embedding_Text_Classification_Pytorch	
Liu et al. [53]	https://github.com/qianliu0708/ToWE	Qian Liu
Zhao et al. [51]	https://github.com/andyweizhao/capsule_text_classification	Wei Zhao
Kumar et al. [79]		
Wang et al. [80]	https://github.com/guoyinwang/LEAM	Guoyin Wang
Aly et al. [32]	https://github.com/uhh-lt/BlurbGenreCollection-HMC	Rami Aly
Zhang et al. [61]	https://github.com/JingqingZ/KG4ZeroShotText	Jingqing Zhang
Chalkidis et al. [62]	https://github.com/iliaschalkidis/lmtc-eurlex57k	Ilias Chalkidis
Pappas et al. [73]	https://github.com/idiap/gile	Nikolaos Pappas
Pan et al. [54]		
Liu et al. [81]		
Guo et al. [56]		
Shi et al. [58]		
Sinoara et al. [82]		
Cai et al. [65]		
Bounabi et al. [85]		
Hu et al. [86]		
Li et al. [74]		

Table 2 continued

Work	Source code repository	Github of authors
Aubaid et al. [83]		
Gupta et al. [84]	https://github.com/vgupta123/SCDV-MS	Vivek Gupta
Meng et al. [66]	https://github.com/yumeng5/LOTClass	Yu Meng
Chang et al. [75]	https://github.com/OctoberChang/X-Transformer	Wei-Cheng Chang
Kim et al. [63]	https://github.com/TeamLab/text-capsule-network	Jaeyoung Kim
Pittaras et al. [55]	https://github.com/npit/nlp-semantic-augmentation/tree/jnle	Nikiforos Pittaras
Liu et al. [59]		
Liu et al. [87]		
Hossain et al. [72]		
Saraswat et al. [89]		
Gallo et al. [60]	https://gitlab.com/nicolalandro/visual_word_embeddings	
Lee et al. [67]	https://github.com/sh0416/oommix	Seonghyeon Lee
Zhang et al. [88]	https://github.com/HeroadZ/KiL	Zhang Cheng
Jiang et al. [68]	http://github.com/kongds/LightXML	Ting Jiang
Moreo et al. [64]	https://github.com/AlexMoreo/word-class-embeddings	Alejandro Moreo

Table 3 Links to datasets used to train and evaluate text classification approaches

Dataset name	Link
20Newsgroups	http://qwone.com/$\sim$jason/20Newsgroups/
5AbstractsGroup	https://github.com/qianliu0708/5AbstractsGroup
AG News	http://groups.di.unipi.it/$\sim$gulli/AG_corpus_of_news_articles.html
BBC Corpus	http://mlg.ucd.ie/datasets/bbc.html
BBCSport labeled	http://mlg.ucd.ie/datasets/bbc.html
BlurbGenreCollection	https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html
CR	https://www.cs.uic.edu/$\sim$liub/FBS/sentiment-analysis.html
CSTR	http://sites.labic.icmc.usp.br/rsinoara/doc-embeddings/
EURLEX 57K	http://nlp.cs.aueb.gr/software_and_datasets/EURLEX57K/
IMDB	http://ai.stanford.edu/$\sim$amaas/data/sentiment/
JRC-Acquis	https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis
MPQA	http://mpqa.cs.pitt.edu/
MR	https://www.cs.cornell.edu/people/pabo/movie-review-data/
MR (2004)	https://www.cs.cornell.edu/people/pabo/movie-review-data/
MR (2005)	https://www.cs.cornell.edu/people/pabo/movie-review-data/
Ohsumed dataset	http://disi.unitn.it/moschitti/corpora.htm
Ohsumed-400	http://sites.labic.icmc.usp.br/rsinoara/doc-embeddings/
R52	https://github.com/yao8839836/text_gcn/tree/master/data/R52
RCV1-V2	http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm
Reuters-21578	http://www.daviddlewis.com/resources/testcollections/reuters21578/
Reuters10	https://www.nltk.org/book/ch02.html
Sogou News	http://www.sogou.com/labs/resource/cs.php
SST	http://nlp.stanford.edu/sentiment
TREC	https://cogcomp.seas.upenn.edu/Data/QA/QC/
WIPO-Gama	https://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/

Table 4 Other datasets used to train and evaluate text classification approaches

Dataset name	References
AAPD	Yang et al. [90]
ACL Anthology Network,	Lai et al. [49]
Ads-1 m	Prabhu et al. [91]
Amazon	Bhatia et al. [71]
Amazon Review	Qiao et al. [70]
Amazon Review Full	Johnson et al. [92]
Amazon Review Polarity	Johnson et al. [92]
Amazon-670K	Jiang et al. [68]
AmazonCat-13k	Jiang et al. [68]
Bengali corpus	Hossain et al. [72]
BioASQ	Nam et al. [93]
ChnSentiCorp-Htl-unba-10000	Li et al. [74]
COPD	Liu et al. [87]
Czech Corpus	Lenc et al. [50]
DBLP (academic paper)	Hu et al. [86]
DBpedia	Zhang et al. [94]
Delicious-Large	Wetzker et al. [95]
Emotions	Kumar et al. [79]
Enron	Liu et al. [59]
EurLEX	Bhatia et al. 2015 [71]
EurLEX-4k	Chang et al. [75]
Fudan Set	Lai et al. [49]
Medical	Liu et al. [59]
MEDLINE citations	Xu et al. [77]
Movielens	Liu et al. [59]
Paper theme data	Li et al. [74]
Ren-CECps	Li et al. [96]
Saraswat corpus	Saraswat et al. [89]
SE-product	Sinoara et al. [82]
SST-2	Zhao et al. [51]
Subjectivity dataset	Zhao et al. [51]
Wiki-500K	Chang et al. [75]
Wiki10	Chang et al. [75]
Wiki10-31K	Zhang et al. [88]
WikiLSHTC	Bhatia et al. [71]
WOS-11967	Kowsari et al. [97]
Yahoo Answer	Johnson et al. [92]
Yelp Review Full	Johnson et al. [92]
Yelp Review Polarity	Johnson et al. [92]

macro F1, precision, recall) achieved by one of the classification methods on each dataset is highlighted in **bold** to allow the reader to better follow and understand the results presented in Tables 5 and 6. It is important to elucidate that the performance scores presented in these tables are taken from the papers that introduced the respective text classification approaches.

5.2 Directions for text classification research

Recent advancements in feature representation and deep learning have propitiated significant progress in text classification, as discussed previously. These advancements include the attention mechanism [98], Transformers [99], BERT [16], and XLNet [76], among others. However, despite these progresses, there are still challenges to be addressed. For the best of our knowledge, current trends and research opportunities for text classification research have not been fully described yet. Thus, based on our experience and what we have identified in our bibliographical review, we point out the following themes as promising directions:

Use of KGs and Knowledge Embedding We envision that knowledge bases such as KGs may contribute to future text classification approaches. Most text classification approaches that exploit embeddings employ only word embeddings, as discussed in Sect. 5. Only a few works use KGs to improve the word embeddings, like [82] and [61]. On the other hand, KGs contain millions of facts describing with precise semantics entities mentioned in the texts to be classified. They can be exploited more efficiently through KG embeddings to improve text classification. Word embeddings and knowledge embeddings are usually trained in independent ways by using different techniques, their respective embeddings are in different vector spaces, hindering their joint use. Nevertheless, these incompatibilities can be overcome by several techniques that combine embedding techniques.

Cost-effective text classification models: Usually, text classification is formulated as a supervised learning problem, where a labeled dataset is used to train a classifier. In practice, training a supervised neural network model requires expensive hardware due to the memory and GPU requirements, and extensive datasets, whose creation may demand a considerable amount of human labor. To meet the computation and storage restrictions of several users and applications, the models have to be compressed. One way it can be done is by building learner models using knowledge distillation or modeling compression techniques [100]. Moreover, the fine-tuning of the existing models for the necessities of an application may alleviate the expensive hardware and dataset size requirements while providing significant improvements.

Effective multi-label classification Traditional multi-label classification methods do not deal adequately with the increasing needs of contemporary big and complex data structures. As a result, there is a critical need for new multi-label learning paradigms, and new trends are emerging [101]. Extreme multi-label classification (XMLC) becomes a developing new line of research that focuses on multi-label problems with a vast number of labels. The existing multi-label classification techniques do not address the XMLC problem due to the prohibitive computational cost. Analyzing all the positive labels to text document poses a challenge in XMLC. An issue in multi-label classification is modeling the interdependencies between labels and features. Existing methods attempt to model the correlations between labels and features. Nevertheless, the statistical properties of these multi-label dependency modelings are less explored, and theoretical analysis is a necessary future research topic.

Table 5 Text classification approaches evaluated with the accuracy score

Work	Dataset 20News groups	5Abstracts Group	ACL Anthology Network	AG News	Amazon Review	Amazon Review Full	Amazon Review Polarity	Amazon-670K	AmazonCat-13K	BBC corpus	BBCSport labeled	Bengali corpus	ChnSenti Corp-Htl-unba-10000	CR	DBLP
Lai et al. 2015	**96.49**		**49.19**												
Jin et al. 2016	83.1														
Kumar et al. 2018															
Qiao et al. 2018				92.8		**60.9**	**95.3**								
Liu et al. 2018	86	**87.2**													
Zhao et al. 2018				92.6										85.1	
Wang et al. 2018				92.45											
Zang et al. 2019	76.7														
Pan et al. 2019															
Liu et al. 2019	84.4														
Guo et al. 2019														**87.5**	
Shi et al. 2019															
Bounabi et al. 2020											**99.1**				
Hu et al. 2020															**76,42**
Li et al. 2020													**89.1**		
Aubaid et al. 2020	90.07														
Meng et al. 2020				86.4	91.6										
Kim et al. 2020	86.74														
Pittaras et al. 2020	78.4									**97.6**					
Gupta et al. 2020	86.19														
Saraswat et al. 2021															

Table 5 continued

Work	Dataset 20News groups	5Abstracts Group	ACL Anthology Network	AG News	Amazon Review	Amazon Review Full	Amazon Review Polarity	Amazon-670K	AmazonCat-13K	BBC corpus	BBCSport labeled	Bengali corpus	ChnSenti Corp-Htl-unba-10000	CR	DBLP
Lee et al. 2021				91.83	**92.94**										
Zhang et al. 2021	87.24			**94.72**											
Jiang et al. 2021								**49.1**	**96.77**						
Hossain et al. 2021												**96.85**			

	DBpedia	Eurlex-4K	Emotions	Fudan Set	IMDB	MPQA	MR (2004)	MR (2005)	MR	MPQA	Ohsumed dataset
Lai et al. 2015				**95.2**							
Jin et al. 2016											
Kumar et al. 2018			**55**								34.4
Qiao et al. 2018	98.9										
Liu et al. 2018					90.8				65.1		
Zhao et al. 2018									82.3		
Wang et al. 2018	99.02										
Zang et al. 2019	85.2										
Pan et al. 2019	92.1										
Liu et al. 2019											
Guo et al. 2019									86.6	**93**	
Shi et al. 2019											

Table 5 continued

	DBpedia	Eurlex-4K	Emotions	Fudan Set	IMDB	MPQA	MR (2004)	MR (2005)	MR	MPQA	Ohsumed dataset		
Bounabi et al. 2020													
Hu et al. 2020									**93.56**				
Li et al. 2020													
Aubaid et al. 2020													
Meng et al. 2020	91.1				86.5								
Kim et al. 2020					89.8		**89**	**81**		90.1			
Pittaras et al. 2020											**43.55**		
Gupta et al. 2020													
Saraswat et al. 2021													
Lee et al. 2021	**99.03**												
Zhang et al. 2021					**94.06**								
Jiang et al. 2021		**87.63**											
Hossain et al. 2021													

	Paper theme data	R52	Reuters-21578	Reuters10	Saraswat corpus	Sogou News	SST	SST-2	Subjectivity dataset	TREC	Wiki-500K	Wiki10-31K	Yahoo Answer	Yelp Review Full	Yelp Review Polarity
Lai et al. 2015							47.21								
Jin et al. 2016			**96.5**												
Kumar et al. 2018															
Qiao et al. 2018						**97.6**							73.7	**64.9**	**96.4**
Liu et al. 2018							**78.8**								
Zhao et al. 2018								**86.8**	93.8	92.8					
Wang et al. 2018													**77.42**	64.09	95.31

Table 5 continued

	Paper theme data	R52	Reuters-21578	Reuters10	Saraswat corpus	Sogou News	SST	SST-2	Subjectivity dataset	TREC	Wiki-500K	Wiki10-31K	Yahoo Answer	Yelp Review Full	Yelp Review Polarity
Zang et al. 2019															
Pan et al. 2019															
Liu et al. 2019			90.2												
Guo et al. 2019									**95.6**	91.9					
Shi et al. 2019						86.53									
Bounabi et al. 2020															
Hu et al. 2020		**97.23**													
Li et al. 2020	**97.7**					97.5									
Aubaid et al. 2020			90.72												
Meng et al. 2020															
Kim et al. 2020				**87.52**						**94.8**					
Pittaras et al. 2020			74.9												
Gupta et al. 2020															
Saraswat et al. 2021					**93**										
Lee et al. 2021													74.13		
Zhang et al. 2021															
Jiang et al. 2021											**77.78**	**89.45**			
Hossain et al. 2021															

Table 6 Text classification approaches evaluated with other metrics instead of accuracy

Metric	Work	Dataset 20News groups	5Abstracts Group	Ads-1 m	AAPD	Amazon	Amazon Cat-13k	BBC Corpus	BioASQ	BlurbGenre Collection	COPD	CSTR
Macro F1	Lai et al. 2015	**96.49**										
	Xu et al. 2016	95										
	Jin et al. 2016	82.7										
	Kumar et al. 2018											
	Sinoara et al. 2019							97.29				**83.44**
	Pittaras et al. 2020	79						**97.6**				
	Liu et al. 2021				**57.1**						**94.4**	
Micro F1	Kumar et al. 2018											
	Chalkidis et al. 2019											
	Sinoara et al. 2019							97.3				**82.63**
	Liu et al. 2019	**83.9**										
	Liu et al. 2021				**72.35**							
	Liu et al. 2021				72.1						**93.5**	
F1	Lenc et al. 2017											
	Liu et al. 2018	85	**87.1**									
	Aly et al. 2019									**74.37**		
	Pappas et al. 2019								**79.85**			
	Liu et al. 2019	**90**										
	Hu et al. 2020											
	Aubaid et al. 2020	70.98										
	Cai et al. 2020				**74.4**							
	Gupta et al. 2020	86.16										
	Moreo et al. 2021	70.7										

Table 6 continued

Metric	Work	Dataset 20News groups	5Abstracts Group	Ads-1 m	AAPD	Amazon	Amazon Cat-13k	BBC Corpus	BioASQ	BlurbGenre Collection	COPD	CSTR
Precision	Lenc et al. 2017											
	Liu et al. 2018	85.5	**86.2**									
	Aly et al. 2019									**77.21**		
	Hu et al. 2020											
	Aubaid et al. 2020	76										
	Cai et al. 2020				**76.8**							
	Gupta et al. 2020	**86.2**										
	Chang et al. 2020						**96.7**					
	Bhatia et al. 2015			**21.84**		**35.05**						
Recall	Lenc et al. 2017											
	Liu et al. 2018	85	**87.1**									
	Aly et al. 2019									**71.73**		
	Hu et al. 2020											
	Aubaid et al. 2020	66.64										
	Cai et al. 2020				**72.2**							
	Gupta et al. 2020	**86.18**										

	Czech Corpus	Delicious-Large	Emotions	Enron	EurLEX	EurLEX-4k	EURLEX 57K	JRC-Acquis	Medical	MEDLINE citations	Movielens	MR
Macro F1												
										50		
			64.9									
Micro F1			**68.2**									
							69.8					
				73.91					**92.22**		**83**	

Table 6 continued

	Czech Corpus	Delicious-Large	Emotions	Enron	EurLEX	EurLEX-4k	EURLEX 57K	JRC-Acquis	Medical	MEDLINE citations	Movielens	MR
F1	**84.19**											
												93.57
								39.7				
Precision	**87.67**											
												93.59
						87.22						
		47.03			**80.17**							
Recall	**83.55**											
												93.56

		Ohsumed-400	Ohsumed dataset	RCV1-V2	Ren-CECps	Reuters-21578	SE-product	Wiki10(30K labels)	Wiki-500k	WikiLSHTC	WIPO-Gama	WOS-11967
Macro F1	Lai et al. 2015											
	Xu et al. 2016											
	Jin et al. 2016					**88.6**						
	Kumar et al. 2018		27.9									
	Sinoara et al. 2019	**38.21**					**95.3**					
	Pittaras et al. 2020		**37.3**			37.8						
	Liu et al. 2021				**58.5**	67.5						
Micro F1	Kumar et al. 2018		46.3									
	Chalkidis et al. 2019											
	Sinoara et al. 2019	**37.96**					**99.26**					

Table 6 continued

		Ohsumed-400	Ohsumed dataset	RCV1-V2	Ren-CECps	Reuters-21578	SE-product	Wiki10(30K labels)	Wiki-500k	WikiLSHTC	WIPO-Gama	WOS-11967
	Liu et al. 2019					90.3						
	Liu et al. 2021		**77.88**									
	Liu et al. 2021				**69.2**	**90.8**						
F1	Lenc et al. 2017					87.59						
	Liu et al. 2018											
	Aly et al. 2019											**81.69**
	Pappas et al. 2019											
	Liu et al. 2019					**93**						
	Hu et al. 2020											
	Aubaid et al. 2020					76.75						
	Cai et al. 2020			**89.9**								
	Gupta et al. 2020					82.71						
	Moreo et al. 2021		**73.1**	69.5		65.2					**57.1**	
Precision	Lenc et al. 2017					91.03						
	Liu et al. 2018											
	Aly et al. 2019											**82.75**
	Hu et al. 2020											
	Aubaid et al. 2020					79						
	Cai et al. 2020			**90.6**								
	Gupta et al. 2020					**95.06**						

Table 6 continued

		Ohsumed-400	Ohsumed dataset	RCV1-V2	Ren-CECps	Reuters-21578	SE-product	Wiki10(30K labels)	Wiki-500k	WikiLSHTC	WIPO-Gama	WOS-11967
	Chang et al. 2020							**88.51**	**77.28**			
	Bhatia et al. 2015							85.54		**55.57**		
Recall	Lenc et al. 2017					86.14						
	Liu et al. 2018											
	Aly et al. 2019											80.67
	Hu et al. 2020					**93.56**						
	Aubaid et al. 2020					75.9						
	Cai et al. 2020			**89.2**								
	Gupta et al. 2020											

6 Conclusions

This paper discussed and compared text classification approaches that use embeddings as feature representations. Our aim was to answer the research questions stated in Sect. 2, by thoroughly analyzing the 38 most relevant papers found and selected in a systematic literature review. We hope it helps to motivate, inspire, understand, and give directions for research in this field.

The selected approaches were compared according with the aspects that we have identified in Sect. 3.2, namely classification problem characteristics like the kind of classification and the datasets used to train and evaluate the proposals, classification techniques employed, data inputs, and data feature representations. Many types of embeddings have been used for representing features for text classification. Word2Vec, GloVe, and BERT are the most used ones in the analyzed papers. The current tendency is to use BERT, probably because it is contextualized and based on the transformer architecture, a recent deep learning architecture that enables model training with significantly larger datasets than other architectures.

The contributions of this work can be stated as follows: (i) appraising the general characteristics of text classification approaches from the literature that exploit a myriad of embedding models, including the ones trained with external information and knowledge sources; (ii) identification of the classification techniques and embedding models that, employed together, have provided the highest performance for distinct corpora; and (iii) provide insights and directions that can be useful for the research community and practitioners in the area of text classification. These contributions can help determine suitable classification techniques and feature representations to be employed in future approaches to tackle text classification challenges.

Regarding the central question of this work, how word embeddings improve text classification, our review shows that text embeddings contribute significantly to the overall performance. These embeddings can capture semantic and syntactic features from word use in corpora. However, though classification models using just text embeddings are more efficient than triplet fact-based models, the information that the former can incorporate is limited. Additional multivariate information with precise semantics, such as concept hierarchies and relations between entities available in a KG, can also be useful to further refine embedding models and improve classification results.

We expect future works to analyze the multilingual aspect of feature representations and the impact of word sense and knowledge embeddings in text classification. Then, certain NLP tasks, such as entity recognition and linking (extraction and disambiguation of mentions to people, companies, locations, events, etc.), could be used for linking textual mentions to specific KG entities, relations and/or their embeddings.

In addition, successful approaches to classify texts in domains explored with more frequency, like news, could be evaluated and adapted to more challenging domains. New approaches could tackle challenges such as multiple terms used for referring to an entity; noisy text (i.e., with typos, grammatical errors, slangs); and lack of contextual information of texts like those of short documents (e.g., social media posts). Lastly, joint classification of multi-modal information, such as text or speech accompanied by images, is a challenge with many potential applications (e.g., detecting diverse categories of hate speech) due to the common use of hypermedia in mass communication nowadays. This challenge may be addressed by exploiting recent or promised breakthroughs in technologies like unified models for multi-modal information and knowledge.

Acknowledgements This study was financed by the Fundação de Amparo á Pesquisa e Inovação do Estado de Santa Catarina—Brasil (FAPESC), by the Print CAPES-UFSC Automation 4.0 Project, and the Brazilian National Laboratory for Scientific Computing (LNCC).

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Gantz J, Reinsel D (2012) The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. IDC iView IDC Anal Future 2007(2012):1–16
2. Altınel B, Ganiz MC (2018) Semantic text classification: a survey of past and recent advances. Inf Process Manag 54(6):1129–1153. https://doi.org/10.1016/j.ipm.2018.08.001
3. Liu W, Wang T (2010) Index-based online text classification for sms spam filtering. J Comput 5(6):844–851
4. Hu W, Du J, Xing Y (2016) Spam filtering by semantics-based text classification. In: Intl. Conf. on advanced computational intelligence (ICACI), pp. 89–94. https://doi.org/10.1109/icaci.2016.7449809. IEEE
5. Dawei W, Alfred R, Obit JH, On CK (2021) A literature review on text classification and sentiment analysis approaches. Computational Science and Technology: 7th ICCST 2020, Pattaya, Thailand, 29–30 August, 2020 724, 305. https://doi.org/10.1007/978-981-33-4069-5_26
6. Melville P, Gryc W, Lawrence RD (2009) Sentiment analysis of blogs by combining lexical knowledge with text classification. In: 15th ACM SIGKDD Intl. Conf. on knowledge discovery and data mining, pp. 1275–1284. https://doi.org/10.1145/1557019.1557156
7. Ahmed H, Traore I, Saad S (2018) Detecting opinion spams and fake news using text classification. Secur Priv 1(1):9
8. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47. https://doi.org/10.1145/505282.505283
9. Deng X, Li Y, Weng J, Zhang J (2019) Feature selection for text classification: a review. Multimed Tools Appl 78(3):3797–3816. https://doi.org/10.1007/s11042-018-6083-5
10. Zha D, Li C (2019) Multi-label dataless text classification with topic modeling. Knowl Inf Syst 61(1):137–160. https://doi.org/10.1007/s10115-018-1280-0
11. Köhn A (2015) What's in an embedding? analyzing word embeddings through multilingual evaluation. In: 2015 Conference on empirical methods in natural language processing, pp. 2067–2073. https://doi.org/10.18653/v1/d15-1246
12. Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. J Mach Learn Res 3:1137–1155. https://doi.org/10.5555/944919.944966
13. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds) Advances in neural information processing systems. Curran Associates Inc, New York
14. Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. In: 2014 Conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543. https://doi.org/10.3115/v1/D14-1162
15. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146. https://doi.org/10.1162/tacl_a_00051
16. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Conf. of the North American Chapter of the ACL, pp. 4171–4186. Association for Computational Linguistics (ACL), s.l
17. Aggarwal CC, Zhai C (2012) A survey of text classification algorithms. In: Mining Text Data, pp. 163–222. Springer, s.l. https://doi.org/10.1007/978-1-4614-3223-4_6
18. Nalini K, Sheela LJ (2014) Survey on text classification. Int J Innov Res Adv Eng 1(6):412–417. https://doi.org/10.1007/978-1-4614-3223-4_6
19. Agarwal B, Mittal N (2014) Text classification using machine learning methods-a survey. In: 2nd intl conf on soft computing for problem solving (SocProS), Dec. 28-30, 2012, pp. 701–709. https://doi.org/10.1007/978-81-322-1602-5_75. Springer

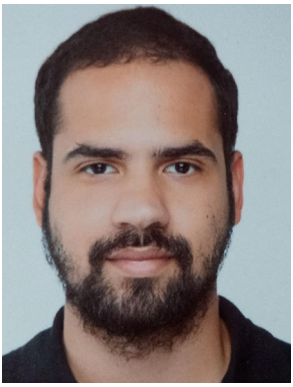
20. Xia L, Luo D, Zhang C, Wu Z (2019) A survey of topic models in text classification. In: 2019 2nd intl conf on artificial intelligence and Big Data (ICAIBD), pp. 244–250 . https://doi.org/10.1109/icaibd.2019.8836970. IEEE
21. Kadhim AI (2019) Survey on supervised machine learning techniques for automatic text classification. Artif Intell Rev 52(1):273–292. https://doi.org/10.1007/s10462-018-09677-1
22. Kowsari K, Jafari Meimandi K, Heidarysafa M, Mendu S, Barnes L, Brown D (2019) Text classification algorithms: a survey. Information 10(4):150. https://doi.org/10.3390/info10040150
23. Zhou Y (2020) A review of text classification based on deep learning. In: 2020 3rd intl conf on geoinformatics and Data Analysis, pp. 132–136. https://doi.org/10.1145/3397056.3397082
24. Yang J, Bai L, Guo Y (2020) A survey of text classification models. In: 2020 2nd intl conf on robotics, intelligent control and artificial intelligence, pp. 327–334. https://doi.org/10.1145/3438872.3439101
25. Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J (2021) Deep learning-based text classification: a comprehensive review. ACM Comput Surv CSUR 54(3):1–40. https://doi.org/10.1145/3439726
26. Stein RA, Jaques PA, Valiati JF (2019) An analysis of hierarchical text classification using word embeddings. Inf Sci 471:216–232. https://doi.org/10.1016/j.ins.2018.09.001
27. Kitchenham B (2004) Procedures for performing systematic reviews. Keele UK Keele Univ 33(2004):1–26
28. Dyba T, Dingsoyr T, Hanssen GK (2007) Applying systematic reviews to diverse study types: an experience report. In: 1st intl. symp. on empirical software engineering and measurement (ESEM), pp. 225–234. https://doi.org/10.1109/esem.2007.59. IEEE
29. Shen W, Wang J, Han J (2015) Entity linking with a knowledge base: issues, techniques, and solutions. IEEE Trans Knowl Data Eng 27(2):443–460. https://doi.org/10.1109/tkde.2014.2327028
30. Oliveira IL, Fileto R, Speck R, Garcia LPF, Moussallem D, Lehmann J (2021) Towards holistic entity linking: survey and directions. Inf Syst 95:101624. https://doi.org/10.1016/j.is.2020.101624
31. Navigli R (2009) Word sense disambiguation: a survey. ACM Comput Surv 10(1145/1459352):1459355
32. Aly R, Remus S, Biemann C (2019) Hierarchical multi-label classification of text with capsule networks. In: 57th annual meeting of the association for computational linguistics: student research workshop, pp. 323–330 . https://doi.org/10.18653/v1/p19-2045
33. Wu L, Yen IE., Xu K, Xu F, Balakrishnan A, Chen P-Y, Ravikumar P, Witbrock MJ (2018) Word mover's embedding: from word2vec to document embedding, 4524–4534. https://doi.org/10.18653/v1/D18-1482
34. Figueiredo F, Rocha L, Couto T, Salles T, Gonçalves MA, Meira W Jr (2011) Word co-occurrence features for text classification. Inf Syst 36(5):843–858. https://doi.org/10.1016/j.is.2011.02.002
35. Grosman JS, Furtado PH, Rodrigues AM, Schardong GG, Barbosa SD, Lopes HC (2020) Eras: improving the quality control in the annotation process for natural language processing tasks. Inf Syst 93:101553. https://doi.org/10.1016/j.is.2020.101553
36. Zhang Y, Jin R, Zhou Z-H (2010) Understanding bag-of-words model: a statistical framework. Int J Mach Learn Cybern 1(1):43–52. https://doi.org/10.1007/s13042-010-0001-0
37. Sparck Jones K (1988) A statistical interpretation of term specificity and its application in retrieval. Taylor Graham Publishing, London
38. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. An introduction to information retrieval. Cambridge University Press, Cambridge
39. Cui P, Wang X, Pei J, Zhu W (2018) A survey on network embedding. IEEE Trans on Knowl Data Eng. https://doi.org/10.1109/TKDE.2018.2849727
40. Lai S, Liu K, He S, Zhao J (2016) How to generate a good word embedding. IEEE Intell Syst 31(6):5–14. https://doi.org/10.1109/mis.2017.2581325
41. Almeida F, Xexéo G (2019) Word embeddings: a survey. arXiv preprint arXiv:1901.09069
42. Bakarov A (2018) A survey of word embeddings evaluation methods. arXiv preprint arXiv:1801.09536
43. Nickel M, Murphy K, Tresp V, Gabrilovich E (2016) A review of relational machine learning for knowledge graphs. IEEE 104(1):11–33. https://doi.org/10.1109/jproc.2015.2483592
44. Wang Y, Cui L, Zhang Y (2019) Using dynamic embeddings to improve static embeddings. CoRR arXiv:1911.02929
45. Tripathi N, Oakes M, Wermter S (2015) A scalable meta-classifier combining search and classification techniques for multi-level text categorization. Int J Comput Intell Appl 14(04):1550020. https://doi.org/10.1142/S1469026815500200
46. Guo N, He Y, Yan C, Liu L, Wang C (2016) Multi-level topical text categorization with wikipedia. In: Proceedings of the 9th iNtl conf on utility and cloud computing. UCC '16, pp. 343–352. Association for Computing Machinery, New York, NY, USA . https://doi.org/10.1145/2996890.3007856. https://doi.org/10.1145/2996890.3007856

47. Aggarwal A, Singh J, Gupta K (2018) A review of different text categorization techniques. Int J Eng Technol UAE 7:11–15
48. Al-Anzi FS, AbuZeina D (2017) A micro-word based approach for arabic sentiment analysis. In: IEEE/ACS 14th Intl. conf on computer systems and applications (AICCSA), pp. 910–914. https://doi.org/10.1109/AICCSA.2017.177
49. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: Twenty-ninth AAAI Conference on Artificial Intelligence, pp. 2267–2273
50. Lenc L, Král P (2017) Word embeddings for multi-label document classification. In: Intl. Conf. Recent Advances in Natural Language Processing, RANLP 2017, pp. 431–437. INCOMA Ltd., Varna, Bulgaria. https://doi.org/10.26615/978-954-452-049-6_057
51. Zhao W, Ye J, Yang M, Lei Z, Zhang S, Zhao Z (2018) Investigating capsule networks with dynamic routing for text classification. In: 2018 conference on empirical methods in natural language processing, pp. 3110–3119. https://doi.org/10.18653/v1/d18-1350
52. Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In: Advances in Neural Information Processing Systems, pp. 3856–3866
53. Liu Q, Huang H-Y, Gao Y, Wei X, Tian Y, Liu L (2018) Task-oriented word embedding for text classification. In: 27th intl conf on computational linguistics, pp. 2023–2032
54. Pan C, Huang J, Gong J, Yuan X (2019) Few-shot transfer learning for text classification with lightweight word embedding based models. IEEE Access 7:53296–53304. https://doi.org/10.1109/access.2019.2911850
55. Pittaras N, Giannakopoulos G, Papadakis G, Karkaletsis V (2021) Text classification with semantically enriched word embeddings. Nat Lang Eng 27(4):391–425. https://doi.org/10.1017/s1351324920000170
56. Guo B, Zhang C, Liu J, Ma X (2019) Improving text classification with weighted word embeddings via a multi-channel textcnn model. Neurocomputing 363:366–374. https://doi.org/10.1016/j.neucom.2019.07.052
57. Kim Y (2014) Convolutional neural networks for sentence classification. In: 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar
58. Shi M, Wang K, Li C (2019) A c-lstm with word embedding model for news text classification. In: 2019 IEEE/ACIS 18th intl conf on computer and information science (ICIS), pp. 253–257. https://doi.org/10.1109/icis46139.2019.8940289. IEEE
59. Liu H, Chen G, Li P, Zhao P, Wu X (2021) Multi-label text classification via joint learning from label embedding and label correlation. Neurocomputing. https://doi.org/10.1016/j.neucom.2021.07.031
60. Gallo I, Nawaz S, Landro N, La Grassa R (2021) Visual word embedding for text classification. Springer, Cham, pp 339–352
61. Zhang J, Lertvittayakumjorn P, Guo Y (2019) Integrating semantic knowledge to tackle zero-shot text classification. In: 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 1031–1040. Association for Computational Linguistics, Minneapolis, Minnesota. https://doi.org/10.18653/v1/n19-1108
62. Chalkidis I, Fergadiotis M, Malakasiotis P, Androutsopoulos I (2019) Large-scale multi-label text classification on EU legislation. In: 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp. 6314–6322. Association for Computational Linguistics, s.l. https://doi.org/10.18653/v1/p19-1636
63. Kim J, Jang S, Park E, Choi S (2020) Text classification using capsules. Neurocomputing 376:214–221. https://doi.org/10.1016/j.neucom.2019.10.033
64. Moreo A, Esuli A, Sebastiani F (2021) Word-class embeddings for multiclass text classification. Data Min Knowl Disc 35(3):911–963. https://doi.org/10.1007/s10618-020-00735-3
65. Cai L, Song Y, Liu T, Zhang K (2020) A hybrid bert model that incorporates label semantics via adjustive attention for multi-label text classification. IEEE Access 8:152183–152192
66. Meng Y, Zhang Y, Huang J, Xiong C, Ji H, Zhang C, Han J (2020) Text classification using label names only: a language model self-training approach. In: EMNLP, pp. 9006–9017. Association for Computational Linguistics, s.l. https://doi.org/10.18653/v1/2020.emnlp-main.724
67. Lee S, Lee D, Yu H (2021) Oommix:out-of-manifold regularization in contextual embedding space for text classification. In: 59th annual meeting of the ACL and the 11th intl joint conf on natural language processing, pp. 590–599. Association for Computational Linguistics (ACL), s.l. https://doi.org/10.18653/v1/2021.acl-long.49
68. Jiang T, Wang D, Sun L, Yang H, Zhao Z, Zhuang F (2021) Lightxml: transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In: The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21), pp. 7987–7994

69. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: 31st intl conf on machine learning (ICML) **4**
70. Qiao C, Huang B, Niu G, Li D, Dong D, He W, Yu D, Wu H (2018) A new method of region embedding for text classification. In: Intl conf on learning representations (Poster), pp. 1–12
71. Bhatia K, Jain H, Kar P, Varma M, Jain P (2015) Sparse local embeddings for extreme multi-label classification. Adv Neural Inf Process Syst 29:730–738
72. Hossain MR, Hoque MM, Sarker IH (2021) Text classification using convolution neural networks with fasttext embedding. In: Abraham A, Hanne T, Castillo O, Gandhi N, Nogueira Rios T, Hong T-P (eds) Hybrid intelligent systems. Springer, Cham, pp 103–113
73. Pappas N, Henderson J (2019) Gile: a generalized input-label embedding for text classification. Trans Assoc Comput Linguist 7:139–155. https://doi.org/10.1162/tacl_a_00259
74. Li Y, Ye M (2020) A text classification model base on region embedding and lstm. In: 2020 6th Intl Conf on Computing and Artificial Intelligence, pp. 152–157. https://doi.org/10.1145/3404555.3404643
75. Chang W-C, Yu H-F, Zhong K, Yang Y, Dhillon IS (2020) Taming pretrained transformers for extreme multi-label text classification. In: 26th ACM SIGKDD Intl Conf on Knowledge Discovery & Data Mining, pp. 3163–3171. https://doi.org/10.1145/3394486.3403368
76. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV (2019) XLNet: generalized autoregressive pretraining for language understanding. Curran Associates Inc., Red Hook
77. Xu H, Dong M, Zhu D, Kotov A, Carcone AI, Naar-King S (2016) Text classification with topic-based word embedding and convolutional neural networks. In: 7th ACM Intl Conf on bioinformatics, computational biology, and health informatics, pp. 88–97
78. Jin P, Zhang Y, Chen X, Xia Y (2016) Bag-of-embeddings for text classification. In: 25th Intl Joint Conf on Artificial Intelligence. IJCAI'16, vol. 16, pp. 2824–2830. AAAI Press, s.l
79. Kumar V, Pujari AK, Padmanabhan V, Sahu SK, Kagita VR (2018) Multi-label classification using hierarchical embedding. Expert Syst Appl 91:263–269. https://doi.org/10.1016/j.eswa.2017.09.020
80. Wang G, Li C, Wang W, Zhang Y, Shen D, Zhang X, Henao R, Carin L (2018) Joint embedding of words and labels for text classification. In: 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp. 2321–2331. Association for Computational Linguistics, Melbourne, Australia. https://doi.org/10.18653/v1/p18-1216
81. Liu W, Liu P, Yang Y, Yi J, Zhu Z (2019) A< word, part of speech> embedding model for text classification. Expert Syst 36(6):12460
82. Sinoara RA, Camacho-Collados J, Rossi RG, Navigli R, Rezende SO (2019) Knowledge-enhanced document embeddings for text classification. Knowl-Based Syst 163:955–971. https://doi.org/10.1016/j.knosys.2018.10.026
83. Aubaid AM, Mishra A (2020) A rule-based approach to embedding techniques for text document classification. Appl Sci 10(11):4009. https://doi.org/10.3390/app10114009
84. Gupta V, Saw A, Nokhiz P, Gupta H, Talukdar P (2020) Improving document classification with multi-sense embeddings. In: 24th European Conference on Artificial Intelligence - ECAI, Santiago de Compostela, Spain, pp. 1–8. IEEE
85. Bounabi M, El Moutaouakil K, Satori K (2020) Neural embedding & hybrid ml models for text classification. In: 2020 1st Intl. Conf. on Innovative Research in Applied Science, Engineering and Technology (IRASET), pp. 1–6 . https://doi.org/10.1109/iraset48871.2020.9092230. IEEE
86. Hu S, He C, Ge B, Liu F (2020) Enhanced word embedding method in text classification. In: 2020 6th Intl Conf on Big Data and Information Analytics (BigDIA), pp. 18–22. https://doi.org/10.1109/bigdia51454.2020.00012. IEEE
87. Liu N, Wang Q, Ren J (2021) Label-embedding bi-directional attentive model for multi-label text classification. Neural Process Lett 53(1):375–389. https://doi.org/10.1007/s11063-020-10411-8
88. Zhang C, Yamana H (2021) Improving text classification using knowledge in labels. In: 2021 IEEE 6th Intl Conf on Big Data Analytics (ICBDA), pp. 193–197. https://doi.org/10.1109/icbda51983.2021.9403092
89. Saraswat A, Abhishek K, Kumar S (2021) Text classification using multilingual sentence embeddings. In: Evolution in Computational Intelligence, pp. 527–536. Springer, s.l
90. Yang P, Sun X, Li W, Ma S, Wu W, Wang H (2018) SGM: sequence generation model for multi-label classification. In: 27th Intl Conf in Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, pp. 3915–3926
91. Prabhu Y, Varma M (2014) Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In: 20th ACM SIGKDD Intl Conf on Knowledge Discovery and Data Mining, pp. 263–272 . https://doi.org/10.1145/2623330.2623651
92. Johnson R, Zhang T (2015) Semi-supervised convolutional neural networks for text categorization via region embedding. Advances Neural Inf Process Syst. Vol 28

93. Nam J, Mencía EL, Fürnkranz J (2016) All-in text: Learning document, label, and word representations jointly. Thirtieth AAAI Conference on Artificial Intelligence. AAAI'16. AAAI Press, Phoenix, Arizona, pp 1948–1954
94. Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. Advances Neural Inf Process Syst. Vol 28
95. Wetzker R, Zimmermann C, Bauckhage C (2008) Analyzing social bookmarking systems: A delicious cookbook. In: ECAI Mining Social Data Workshop, pp. 26–30
96. Li J, Ren F (2011) Creating a chinese emotion lexicon based on corpus ren-cecps. In: 2011 IEEE Intl Conf on Cloud Computing and Intelligence Systems, pp. 80–84. https://doi.org/10.1109/ccis.2011.6045036. IEEE
97. Kowsari K, Brown DE, Heidarysafa M, Meimandi KJ, Gerber MS, Barnes LE (2017) Hdltex: Hierarchical deep learning for text classification. In: 2017 16th IEEE Intl Conf on Machine Learning and Applications (ICMLA), pp. 364–371. https://doi.org/10.1109/icmla.2017.0-134. IEEE
98. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. CoRR arXiv:1409.0473
99. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Systems. Vol. 30
100. Wang W, Wei F, Dong L, Bao H, Yang N, Zhou M (2020) Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. Adv Neural Inf Process Syst 33:5776–5788
101. Liu W, Wang H, Shen X, Tsang I (2021) The emerging trends of multi-label learning. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/tpami.2021.3119334

Liliane Soares da Costa is a Ph.D. student in the Department of Informatics and Statistics at The Federal University of Santa Catarina, working under the supervision of Prof Renato Fileto. Liliane completed her master's degree with distinction in Computer Science at The Federal University of Viçosa. Her thesis research applies machine learning and deep learning in the text classification domain. Her research interests include machine learning and computer vision.



Italo Lopes Oliveira is a data scientist in Jusbrasil, working with NLP in Brazilian legal documents. Ph.D. in Computer Science from the Federal University of Santa Catarina. His research interests include machine learning, NLP, and the semantic web.

Renato Fileto has a Bachelor's degree in Computer Science from the Federal University of Uberlândia, Master's and Doctorate degrees in Computer Science from Campinas State University with an internship at Georgia Institute of Technology, and a Post-Doctorate from the University of São Paulo. Since 2006, he is a permanent professor at the Department of Informatics and Statistics (INE) of Santa Catarina Federal University (UFSC), in Florianópolis-SC, Brazil. His research area is databases and data science, with a focus on semantics and contexts for data analytics.