A survey of text classification: problem statement, latest methods and popular datasets

Siyu Tian[1, a], Xinyao Huang[2, b]

[1]School of Literature and Journalism, Sichuan University, Chengdu, China.

[2]School of Social and Public Administration，East China University of Science and Technology, Shanghai, China.

[a]19003098@mail.ecust.edu.cn; [b]1007002186@qq.com

Abstract. Considering the important role text classification plays in natural language processing tasks, improving the accuracy and efficiency of text classification has been a priority in recent work. In this paper, we focus on the latest text classification methods and sort them into three categories: embedding methods, language models, and various neural networks. We summarize the state of current research and the insufficiencies which may be directions for future study.

Keywords: Text classification, Document Classification, Text Categorization.

1. Introduction

Text classification involves the automatic classification and labelling of a large number of texts based on certain standards. As the basis of many downstream tasks, such as topic classification, sentiment analysis, and so forth, text classification is predominantly used in natural language processing (NLP) tasks.

With the exponential increase in quantity of text information, the scale of the corpus is expanding rapidly, bringing with it a large computation burden. Concurrently, each text consists of multiple sentences which may express different meanings and hard-to-capture keywords that highlight their thematic or sentimental tendencies from the whole text. Factors such as these create particular challenges for text classification. Accordingly, researchers are committed to adopting various methods to improve the efficiency and accuracy of text classification, especially from the perspective of semantics.

We organises research conducted to optimise text classification since 2018, and divides them into three categories according to the optimisation method used: embedding methods, language models, and neural network structures. Embedding is a necessary way to convert unstructured information into structured information; it can be divided into word embedding, document embedding, label embedding, and contextual embedding (according to the different levels and contents of embedding methods). The language models aim to calculate the probability of a sentence and is divided into pre-trained models and non-pre-trained models, respectively. The mainstream of current text classification task innovation is to modify and innovate neural network structures, such as CNN, RNN, Transformer, Attention Mechanism, and the like.

This survey is organised as follows. In Section 2, we introduce embedding methods such as optimising word embedding, context embedding, and others, so as to improve the representation of documents. In Section 3, we discuss the use of language models, primarily in combination with pre-trained models. In Section 4, we describe methods that modify and innovate multiple neural network structures. We conclude the survey in Section 5 by analysing the current research priorities and future directions for researchers to consider.

Table 1. Classification

Category	Method	Architecture	Dataset
eural Networks	BIGBIRD	Attention	IMDb, Arxiv, Patents, Hyperpartisan, Yelp-5
	NABoE	Attention	20 Newsgroups, R8
	Hierarchical approach	Attention	WOS, DBpedia
	MPAD	Attention	Reut., BBC, Pol., Subj., MPQA, IMDB, TREC, SST-1, SST-2, Yelp'13
	HAHNN	Attention	Yelp2018, IMDb
	Capsule Networks with Dynamic Routing	Capsule	MR, SST-2, Subj, TREC, CR, AG's news
	DS-Caps	Capsule	MR(2004), MR(2005), Reuters, TREC-QA, MPQA, IMDB
	MBCH	CNN	CR, IMDB, MPQA, Subj
	SingleCNN-SPE	CNN	IMDB, MR, MPQA, TREC, Reuters, 20news
	VDCNN	CNN	AG's News, Yelp Polarity, Yelp Full
	TextING	GNN	MR, R8, R52, Ohsumed
	TextGCN	GNN	20Newsgroups, Ohsumed,, R52, R8, Reuyers21578, MR
	Text Level Graph Neural Network	GNN	R8, R52, Ohsumed
	DRNN	RNN	AG, DBpedia, Yelp Polarity
	Seq2CNN	RNN	AG's News, DBPedia, Yahoo Answers
	BiLSTM	RNN	Reuters-21578, AAPD, IMDB, Yelp2014
	Hi-Transformer	Transformer	Amazon, IMDB, MIND
	ERNIE-DOC	Transformer	WikiText-103
	X-Transformer	Transformer	Eurlex-4K, Wiki10-31k, AmazonCat-13K, Wiki-500K
	More Identifiable yet Equally Performant Transformers	Transformer	IMDB, TREC, SST, SNLI, Yelp, DBPedia, Sogou News, AG News, Yahoo! Answers, Amazon Reviews
Language Models	BERT with fusing Label Embedding	Pre-trained	AGNews, DBPedia, Yahoo!Answers Topic, IMDb, Yelp Review Full, Yelp Review Polarity
	CoLAKE	Pre-trained	Wikidata5M
	BAE	Pre-trained	Amazon, Yelp, IMDB, MR, MPQA, Subj, TREC
	BERTgrid	Pre-trained	N/A
	ELECTRA	Pre-trained	CoLA, SST, MRPC, STS, QQP, MNLI, QNLI, RTE
	DocBERT	Pre-trained	Reuter, AAPD, IMDB, Yelp'14
	ULMFiT	Un-pre-trained	IMDb, TREC-6, AG
Embedding Methods	WME	Word Embedding	BBCSPOCT, TWITTER, RECIPE, OHSUMED, CLASSIC, REUTERS, AMAZON, 20news, RECIPE_L
	LEAM	Label Embedding	AGNews, Yelp Binary, Yelp Full,DBPedia,Yahoo
	Word and Document Embedding with vMF-Mixture Priors	Word Embedding	20NG, OHS, TechTC, Reu
	CWC	Word Embedding	AG, DBP, Yah.A, Sogou, Yelp.P, Welp.F, Amz.P, Amz.F
	task-oriented word embedding	Word Embedding	20NewsGroup, 5AbstractsGroup, IMDB, MR, SST
	WMD	Word Embedding	BBCSPORT, TWITTER, RECIPE, OHSUMED, CLASSIC, REUTERS, Amazon, 20News

VLAWE	Word Embedding	Movie Review
EXAM	Word Embedding	Amazon Review Polarity, Amazon Review Full, AG's News, Yahoo!Answers, DBPedia
Smaller Text Classifiers with Discriminative Cluster Embeddings	Word Embedding	AG News
VMASK	Word Embedding	IMDB, SST-1, SST-2, Yelp, AG News, TREC, Subj
Document Embeddings trained with Cosine Similarity	Document Embedding	IMDB
Knowledge-enhanced document embeddings	Document Embedding	BBC, SE-product, BS-topic, CSTR, Ohsumed-400, SE-polarity, SE-product-polarity, BS-semantic, BS-topic-semantic
a tf-idf weighted document vector embedding	Document Embedding	N/A
SCDV	Document Embedding	Reuters-21578
P-SIF	Document Embedding	SemEval(2012-2017)
SCDV-MS	Document Embedding	20NewsGroup
General Purpose Text Embeddings from Pre-trained Language Models	Document Embedding	MNLI, QNLI, QQP, RTE, SST-2, MRPC, CoLA, AG-news, Amazon-5, Amazon-2, Yelp-5, Yelp-2, DBpedia
SDDE	Document Embedding	IMDB, Yelp P., AG's News, DBPedia
A Cluster-based Approach in Contextual Embedding Space	Contextual Embedding	STS 2012-2016, SICK-Relatedness, STS-B
Out-of-manifold Regularization in Contextual Embedding Space	Contextual Embedding	AG News, Amazon Review, Yahoo Answer, DBpedia
Fusing Label Embedding into BERT	Label Embedding	AGNews
LCM	Label Embedding	20NG, AG's News, DBPedia, FDCNews, THUCNews

2. Embedding Methods

2.1 Word Embedding

Word embedding is a method of converting words into digital vectors in order to represent where unstructured information is transformed into structured information. To facilitate text classification, researchers try to optimise word embedding to form a better document representation.

The vector representation of word meaning is primarily determined by the context of words. Therefore, it is of great importance to grasp the context of word vectors. Since words appearing in similar contexts have similar word vectors, it is possible to organise word vectors with the same topic through clustering. To learn context word vectors organised in clusters, Schockaert and Jameel (2019) imposed von Mises-Fisher (vMF) distributions on the content word vectors, as vMF is appropriate to model clusters in directional data. As a result, the content word vectors were clustered. When the words in the document are regarded as context word vectors, they can be further applied to the document representation.

Although contextual information is important to some extent, task-specific features will also be significant when seeking to ensure effective document classification. In practical applications of text classification, the lack of task-specific features contributes to the performance of text classification. Therefore, Liu et al. (2018) proposed a task-oriented word embedding method through the regularization of the distribution of words to create a clear classification boundary. To learn task-

specific word representation, Chen and Ji (2020) turned to the help of the Variational Word Mask (VMASK), which comprises three neural text classifiers to improve the interpretability of model predictions.

Generally, the drawback of document embedding methods such as bag of words and term frequency-inverse document frequency is that they cannot capture the distance between words and documents. To measure the distance between documents, Kusner et al. (2015) presented a metric called word mover's distance (WMD). The distance between two documents is the minimum distance between the words embedded by word2vec in one document and the point cloud in the other document. WMD has proven to show very strong interpretability and accuracy. With the application of WMD, distance measures between inputs have often proved to outperform feature embedding for structured inputs. Wu et al. (2018) constructed an efficient distance-based model for classification tasks, which generates novel positive definite kernels from the dissimilarity measure between input objects, thus constituting a feature of embedding. This approach is far superior to other distance-based algorithms. Based on D2KE, Wu et al. (2018) further improved it and proposed a document embedding approach called the Word Mover's Embedding (WME) from pre-trained word embeddings, which is more efficient due to constructing a text kernel using WMD to learn vector representations for texts in order to accelerate the computation of WMD and its variants.

Since word embeddings necessitate a large number of parameters, it is hard to sustain such large storage and memory footprints. Accordingly, efforts have been made to reduce the number of parameters for word embeddings while maintaining the models' performance. Chen and Gimpel (2019) introduce an alternative parameterization by using clusters in an end-to-end manner for word embeddings, as a result of which clusters can make use of parameters more efficiently by avoiding learning excessive embedding vectors. Ren and Lu (2018) adopted a Compositional Weighted Coding (CWC) embedding method that generates word embedding by all code-word vectors in each codebook. They also proposed a routing algorithm, applying capsule network and K-means clustering to model the relationships between word embeddings.

There are multiple and increasingly complex ways to convert word vectors into document-level representations. To achieve high quality results in a more straightforward way, Ionescu and Butnaru (2019) proposed the Vector of Locally-Aggerated Word Embeddings (VLAWE) method inspired by the Vector of Locally-Aggregated Descriptors (VLAD) method, usually applied in image classification, thus becoming the first to adopt VLAD in the text-domain. The VLAWE representation is computed by cumulating the differences between code-word vectors and word vectors that have been clustered using k-means. Due to its applicability to new words, it can be used for performing multiple text classifications.

Deep neural network-based methods of text classification usually generate word-level representations from the whole input text, after which they transform the word-level representations into text-level representations through aggregation operations. Although methods based on deep neural networks are widely employed in the process of text classification, they usually ignore the fine-grained classification clues. To solve this problem, Du et al. (2018) propose a framework named the Explicit interAction Model (EXAM) to compute the word-level interacting signals for the text classification. An interaction matrix is formed using the word-level representation and interpreted as a text representation. This reflects the matching degree of a word and a class which may signify a word-level representation. Consequently, fine-grained word-level information can be taken into account by treating this interaction matrix as a text-level representation.

2.2 Label Embedding

Label embedding requires a model to process label-related information and input features simultaneously. Learning the representation of text not only incurs lower operation costs, but can also capture the semantics of word sequences, making Label embedding an important factor influencing the performance of the model. However, to date, research focused on label embedding for text or document representations remains insufficient.

Through regarding text classification as an issue of label-word joint embedding, Wang et al. (2018) proposed an attention-based label embedding framework called the Label Embedding Attentive Model (LEAM) that learns from the latent space which the word and label are jointly embedded in; text representations are constructed by measuring the text-label compatibility.

Text classification models previously relied on neural networks that usually apply a one-hot label vector to represent labels, making it difficult to recognise some labels, creating the so-called Label Confusion Problem (LCP). To mitigate the LCP, Guo et al. (2020) proposed a label embedding method named the Label Confusion Model (LCM) that is able to grasp the relationships between labels and instances by calculating their similarity. They employ a simulated label distribution as an alternative to a one-hot label vector to improve the classification performance.

2.3 Contextual Embedding

Contextual embedding characterises each word according to its context, thereby achieving a better word representation both syntactically and semantically, also making it transferrable across diverse languages (Liu, Kusner & Blunsom, 2020).

The degeneration problem in contextual word representations means that they are distributed in a narrow cone, with even irrelevant words showing high cosine similarities, which further affects the embedding space and the encoded knowledge. As the contextual word representation adopts the cluster structure, Chen and Gimpel (2019) sought to address the issue through a cluster-based method that clusters embeddings and searches for principal components indicating the main direction of each cluster.

2.4 Document Embedding

Document embedding aims to provide vector representations of documents as input features. Due to the growing need for applying understandable document representation in classification tasks, some methods directly optimise document embedding to improve text classification, without intermediate steps such as word or label embedding,

The large number of parameters required by the pre-training model carry a huge computation burden. Therefore, Du et al. (2020) proposed a shared computation method that uses a text encoder for pre-training for different specific tasks to amortize the computational cost during the inference. This approach proves to be efficient in large-scale pre-trained models, and is able to perform multiple tasks at the same time.

The weighted averaging of word vectors in order to obtain representations of sentences can only be used with a single sentence. For texts containing many sentences, it is not applicable as it is unable to understand the semantics of the whole document. Representing documents in dense, low-dimensional spaces is challenging. Mekala et al. (2017) proposed the Sparse Composite Document Vector (SCDV) to improve the performance of text representation, syntactic and semantic information learned by latent topic models. Nonetheless, Gupta et al. (2019) have contended that SCDV has some drawbacks, such as ignoring multiple meanings of words and suffering from high dimensionality. To optimise SCDV, they propose an SCDV-MS multi-sense embedding method. This eliminates the fuzzification of multiple meanings of words as an unintended consequence of context words and adds sparsity to the fuzzy word cluster assignments that are helpful in lower-dimensional manifold representation.

To alleviate the inefficiencies of obtaining document embeddings when operating on word embeddings where the words in the document usually come from multiple topics, Gupta et al. (2020) presented P-SIF: a simple partitioned word averaging model designed to learn topic-based word vectors that are consequently adopted to represent the topical structure of documents.

The majority of existing document representation methods only focus on the internal information of the text, and predict the context of the target word without considering the relationships between different documents within the same corpus. To combat this, Chen et al. (2019) proposed a framework called the Self-discriminative Document Embedding (SDDE) model, trained to enlarge the distance

between a sentence and the document to capture inter-document relationships among training documents.

Cosine similarity can reflect the similarity of documents, and so can be used to distinguish between different types of documents. The smaller the cosine similarity of two documents, the better. To maximize the cosine similarity, Thongtan and Phienthrakul (2019) used cosine similarity to train document embeddings, instead of using dot-product through improvements made in previous works such as Document Vector by predicting n-grams etc.

3. Language Models

Efforts have been made to improve text classification tasks by designing novel language models, mostly based on pre-trained models. One of the most noted ones is BERT; it employs a masked language model (MLM) that enables the generation of deep bidirectional language representations. BERT can be used in multiple types of natural language processing, such as question answering and language inference.

Masked language models like BERT learn only a small proportion of tokens in each sentence, leading to high computation costs. Adhikari et al. (2019) were the first to apply BERT in document classification. Initially, they fine-tuned BERT to make it suitable for document classification. Next, to reduce the burden of computation, the researchers used the knowledge distillation method to transfer BERT into a simpler neural network model, which also exhibits similar performance levels to BERT. Moreover, Clark et al. (2020) produced Replaced Token Detection, which differs from the masked language model in that this discriminative model corrupts the input by using samples from a proposal distribution. It learns and corrupts all input tokens, instead of only the small masked subset, resulting in the enhancement of operational efficiency. Tests have shown that Replaced Token Detection outperforms most BERT-based models.

To enhance the performance of BERT in text classification, Xiong et al. (2021) proposed the idea of a contextual representation that embeds text and labels learned in the same latent space at the same time, while using BERT's input structure and attention mechanism to learn the relationship between label embeddings and text embeddings. This enables BERT to perform well on small and medium-sized benchmarks.

To improve the robustness of the pre-trained model, researchers adopted adversarial attacks to increase text noise in order to make the pre-trained model more stable and thus perform better. While improving the accuracy of machine learning models, security cannot be overlooked, for example in terms of facing adversarial attacks. Therefore, it is crucial to generate adversarial examples, the difficulty of which lies in the backpropagation caused by the disturbance from the embedding space and the token space. The token replacement strategy can compensate for the issue of the semantics of the whole sentences being ignored by rule-based synonym replacement strategies to generate adversarial examples. For instance, BAE can be applied to generate adversarial examples (Garg and Ramakrishnan, 2020). BAE initially replaces and inserts tokens in the original document, after which it generates masked tokens in the original text according to the importance of tokens in the text classification by the BERT masked language model. Finally, it creates adversarial examples syntactically and semantically.

In NLP tasks, the layouts of documents sometimes contain important semantic information. A lack of layouts, usually described as a sequence of words, can influence the performance of downstream tasks. On the other hand, the computer vision methods that apply various procedures to work at the raw document pixel level are not feasible because the textual-level information is still learned first. Therefore, the NLP method and the CV method can be combined. A typical example of this is the combination of BERTgrid with CNN for document embedding (Denk and Reisswig, 2019). Pre-trained on a large set of unlabelled documents, this method generates word embeddings with context vectors in a 2D grid, performing well in document header field extraction.

Optimising neural networks with pre-trained models will invariably lead to issues with over-parameterization, especially when considering models with small training data due to the large pre-trained weight. In line with to the fact that only low-dimensional subspaces are suitable for fine-tuning pre-trained models, this problem could be ameliorated by regularizing the application of out-of-manifold embeddings that cannot be accessed by words in order to fine-tune task-specific neural networks (Lee, Lee, & Yu, 2021).

In addition to the above BERT-related methods, some methods simply optimise the non-pre-trained language model. Inductive learning performs extremely well in CV, but lacks task-specific modifications for applications in NLP. A Universal Language Model Fine-tuning (ULMFiT) (Howard & Ruder, 2018) is a transfer learning method that has strong applicability and can be implemented with any NLP task. In combination with several new methods of fine-tuning the model proposed in the article, it is effective in avoiding catastrophic forgetting during the process.

4. Neural Networks

Extensively used in various fields of natural language processing, many methods improve the performance of text classification thanks to the optimisation of various neural networks to represent documents more effectively.

4.1 Convolutional Neural Network

CNNs (Convolutional Neural Networks) have always shown high-quality performance in text classification tasks. However, CNNs display some drawbacks in that they overlook the local order of words.

A trend in the development of neural networks is that to improve accuracy in different tasks, the number of layers of neural network models is increasing, resulting in a very high quantity of parameters. To reduce the necessity for parameters while maintaining model performance as much as possible, Duque et al. (2019) combined Temporal Depthwise Separable Convolutional and Global Average Pooling techniques to modify the Very Deep Convolutional Neural Network, as a result of which they proposed the Squeezed Very Deep Convolutional Neural Networks (SVDCNN), which are suitable for mobile platforms.

As very deep neural networks encounter the problems of vanishing and exploding gradients, the Highway network is applied alongside CNN to overcome these issues (Rezaeinia, Ghodsi, & Rahmani, 2018). This form of architecture is called Multiple Block Convolutional Highways (MBCH). Despite the shallowness and simplicity of this method, it can still improve the accuracy of text classification.

4.2 Attention Mechanism

Attention mechanisms that focus on important information were first used in image processing and later played a vital role in the field of natural language processing. The innovation of the attention mechanisms lies in the fact that it has strong interpretability and can provide advantageous information. Attention mechanisms are commonly combined with neural network structures to perform better in completing various tasks. Sinha et al. (2018) attempted to solve the text classification problem resulting from the increasing number of documents and adopted a variant attention mechanism that computes attention weights conditioned on the parent category embedding in order to generate dynamic document representation, forming an end-to-end deep neural-based classifier.

Since many models for text classification fail to take into account the structure of the article as well as the critical words and sentences, Abreu et al. (2019) proposed a model called Hierarchical Attentional Hybrid Neural Networks (HAHNN), based on the word and sentence levels. CNN can extract abundant features through hierarchical representation, with the attention mechanism improving the accuracy of text classification.

Multiple text classification models based on neural networks only grasp the semantics of documents at the word level by embedding composite functions into words. As a result of this, some models use document-relevant entities in a knowledge base (KB) to better represent documents, since entities indicate semantic information of accuracy. Subsequently, centring on how to associate documents and document-related entities, Yamada and Shindo (2019) developed a neural network model that combines dictionary-based entity detection and attention mechanism to focus on document-related entities in documents.

4.3 Capsule Network

A Capsule Network, which uses neuron vectors to replace the single neuron node found in traditional neural networks and trains through dynamic routing, can reduce the disadvantages of CNNs. For example, CNNs will lose some local information while obtaining invariance through pooling operations, and it is challenging to explain the positional relationship between the part and the whole.

Models that rely on pre-trained contextual embeddings often require a large corpus to perform effectively in the downstream tasks of text classification. Researchers sought help from the capsule network, which is a suitable alternative to replace models relying on contextual embeddings. However, existing capsule networks do not take into consideration the features of text sequences. To solve this problem, Demotte and Ranathunga (2021) proposed a new dual state capsule (DS caps) that combines two types of States and capsule layers at word and sentence level to process and learn context-level data in order. In the first work that applies capsule networks to text modelling, Zhao et al. (2018) proposed three methods to optimise the dynamic routing process, aiming to alleviate the interference of noisy capsules.

4.4 Graph Neural Network

Graph neural networks (GNNs) are efficient in processing data that exists in graph structures, and able to handle complex structures of texts. The first method to treat the entire corpus as a graph was presented by Yao, Mao, and Luo (2018). After constructing the graph based on word co-occurrence and document word relations, they applied GNN to learn word and document embeddings.

The primary framework of GNNs is the framework of Message Passing (MP). However, few applications adopt the MP framework to represent documents. Consequently, the Message Passing Attention network for Document understanding (MPAD) (Nikolentzos, Tixier, & Vazirgiannis, 2019) treats documents as word co-occurrence networks, thereby focusing on word order and word-to-word relationships.

As the GNN model typically builds only one graph for the entire corpus, it must to use a large connection window, thus leading to high memory consumption. In addition, graphs are corpus-dependent and cannot be modified after training. Aiming to resolve these problems, Huang et al. (2019) introduced a new GNN model that generates a document-level graph for each input document and uses a smaller window to generate the graph.

These GNN models possess the drawback of setting little store by the relationship between words in each document. What is more, it is difficult to use GNN for inductive learning because it adopts a global structure. For this reason, Zhang et al. (2020) trained TextING, a GNN model that can reflect the word-word relationship at the text level, and similar to the work of Huang et al. (2019), each document also has its own graph. TextING is capable of generating word embeddings for new documents in an inductive case.

4.5 Recurrent Neural Network

RNN (Recurrent Neural Network) is a neural network processing sequence data that is better suited to handling time-series-related tasks such as machine translation. Although RNNs are good at modeling sequences, they are inadequate when it comes to extracting key features. Therefore, Wang (2018) used max pooling to extract key information and added position-invariance to the RNN,

creating the Disconnected Recurrent Neural Network (DRNN). Overly complex neural network structures, usually favoured by researchers, prompted Adhikari et al. (2019) to propose a simple LSTM model using appropriate regularization methods such as embedding dropout, weight dropping, and temporal averaging.

5. Transformer

Transformer-based models are now common in natural language processing domains, especially with regard to many sequence-based tasks. The core of the Transformer lies in the self-attention mechanism that learns word dependencies without considering the distance between words. However, a consequence of this is that the difficulty of the operation is closely related to the length and scale of the text. Therefore, modeling long documents with transformers is difficult. To this end, Ding et al. (2020) proposed a transformer-based pre-trained language model called ERNIE-Doc (A Retrospective Long-Document Modeling Transformer), equipped with two mechanisms to understand the overall semantics of the document. Wu et al. (2021) designed a hierarchical interactive Transformer (Hi-Transformer) that initially learns sentence representations, followed by document representations. Hi-Transformer can effectively reduce the complexity of computing and improve the accuracy of understanding the documents' contents.

The extreme multi-label text classification problem (XMC), aiming to select the labels most relevant to the input text from a large label collection, is a great challenge in NLP tasks. Using the Transformer model to solve XMC results in poor performance due to the vast output space and label sparsity issues. Therefore, Chang et al. (2020) proposed X-Transformer, the first method specifically targeting the XMC problem by fine-tuning the transformer model through a Semantic Label Indexing component, a Deep Neural Matching component, and an Ensemble Ranking component.

6. Conclusion

Given the crucial role that text classification plays in the field of natural language processing and the obstacles it faces in performing better, we organise the latest work on improving text classification performance and divide it into three categories according to its contents, based on which we can summarize current research trends. First, the use of neural network structures is becoming increasingly diverse besides the basic CNN, RNN, and others. Additionally, the popularity of pre-trained models, especially BERT, has led to their widespread use. Finally, researchers are excavating richer semantic information, which is conducive to being more applicable in practical scenarios and uses. At the same time, it is worth mentioning that there are still not many methods for directly optimising document representation, which may be a valuable direction of further work.

References

[1] Abreu, J., Fred, L., Macêdo, D., & Zanchettin, C. (2019). Hierarchical Attentional Hybrid Neural Networks for Document Classification. https://doi.org/10.1007/978-3-030-30493-5_39.

[2] Adhikari, A., Ram, A., Tang, R., & Lin, J. (2019). DocBERT: BERT for Document Classification. http://arxiv.org/abs/1904.08398.

[3] Adhikari, A., Ram, A., Tang, R., Lin, J., & Cheriton, D. R. (2019). Rethinking Complex Neural Network Architectures for Document Classification (pp. 4046–4051). https://github.com/lancopku/SGM.

[4] Benballa, M., Collet, S., & Picot-Clemente, R. (2020). Saagie at Semeval-2019 Task 5: From Universal Text Embeddings and Classical Features to Domain-specific Text Classification (pp. 469–475). https://github.com/shivam5992/.

[5] Chang, W. C., Yu, H. F., Zhong, K., Yang, Y., & Dhillon, I. S. (2020). Taming Pretrained Transformers for Extreme Multi-label Text Classification. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 3163–3171. https://doi.org/10.1145/3394486.3403368.

[6] Chen, H., & Ji, Y. (2020). Learning Variational Word Masks to Improve the Interpretability of Neural Text Classifiers. http://arxiv.org/abs/2010.00667.

[7] Chen, H.-Y., Hu, C.-H., Wehbe, L., & Lin, S.-D. (2019). Self-Discriminative Learning for Unsupervised Document Embedding (pp. 2465–2474). Association for Computational Linguistics.

[8] Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. http://arxiv.org/abs/2003.10555.

[9] Demotte, P., & Ranathunga, S. (2021). Dual-State Capsule Networks for Text Classification. http://arxiv.org/abs/2109.04762.

[10] Denk, T. I., & Reisswig, C. (2019). BERTgrid: Contextualized Embedding for 2D Document Representation and Understanding. http://arxiv.org/abs/1909.04948.

[11] Ding, S., Shang, J., Wang, S., Sun, Y., Tian, H., Wu, H., & Wang, H. (2020). ERNIE-Doc: A Retrospective Long-Document Modeling Transformer. http://arxiv.org/abs/2012.15688.

[12] Du, C., Chin, Z., Feng, F., Zhu, L., Gan, T., & Nie, L. (2018). Explicit Interaction Model towards Text Classification. http://arxiv.org/abs/1811.09386.

[13] Du, J., Ott, M., Li, H., Zhou, X., & Stoyanov, V. (2020). General Purpose Text Embeddings from Pre-trained Language Models for Scalable Inference. http://arxiv.org/abs/2004.14287.

[14] Duque, A. B., Santos, L. L. J., Macêdo, D., & Zanchettin, C. (2019). Squeezed Very Deep Convolutional Neural Networks for Text Classification. https://doi.org/10.1007/978-3-030-30487-4_16.

[15] Garg, S., & Ramakrishnan, G. (2020). BAE: BERT-based Adversarial Examples for Text Classification. http://arxiv.org/abs/2004.01970.

[16] Guo, B., Han, S., Han, X., Huang, H., & Lu, T. (2020). Label Confusion Learning to Enhance Text Classification Models. http://arxiv.org/abs/2012.04987.

[17] Gupta, V., Saw, A., Nokhiz, P., Netrapalli, P., Rai, P., & Talukdar, P. (2020). P-SIF: Document Embeddings Using Partition Averaging. https://github.com/vgupta123/P-SIF.

[18] Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. http://arxiv.org/abs/1801.06146.

[19] Ionescu, R. T., & Butnaru, A. M. (2019). Vector of Locally-Aggregated Word Embeddings (VLAWE): A Novel Document-level Representation. http://arxiv.org/abs/1902.08850.

[20] Kant, N., Puri, R., Yakovenko, N., & Catanzaro, B. (2018). Practical Text Classification With Large Pre-Trained Language Models. http://arxiv.org/abs/1812.01207.

[21] Kim, T., & Yang, J. (2018). Abstractive Text Classification Using Sequence-to-convolution Neural Networks. http://arxiv.org/abs/1805.07745.

[22] Kowsari, K., Heidarysafa, M., Brown, D. E., Meimandi, K. J., & Barnes, L. E. (2018). RMDL: Random multimodel deep learning for classification. ACM International Conference Proceeding Series, 19–28. https://doi.org/10.1145/3206098.3206111.

[23] Kusner, M. J., Sun, Y., Kolkin, N. I., & Weinberger, K. Q. (2015). From Word Embeddings To Document Distances. 10.

[24] Lee, S., Lee, D., & Yu, H. (2021). Out-of-Manifold Regularization in Contextual Embedding Space for Text Classification. http://arxiv.org/abs/2105.06750.

[25] Liu, Q., Huang, H., Gao, Y., Wei, X., Tian, Y., & Liu, L. (2018). Task-oriented Word Embedding for Text Classification (pp. 2023–2032).

[26] Luo, D., Cheng, W., Ni, J., Yu, W., Zhang, X., Zong, B., Liu, Y., Chen, Z., Song, D., Chen, H., & Zhang, X. (2021). Unsupervised Document Embedding via Contrastive Augmentation. http://arxiv.org/abs/2103.14542.

[27] Nikolentzos, G., Tixier, A. J.-P., & Vazirgiannis, M. (2019). Message Passing Attention Networks for Document Understanding. http://arxiv.org/abs/1908.06267.

[28] Oh, B.-D., & Kim, Y.-S. (2020). Lightweight Text Classifier using Sinusoidal Positional Encoding.

[29] Pang, B., & Wu, Y. N. (2021). Latent Space Energy-Based Model of Symbol-Vector Coupling for Text Generation and Classification. http://arxiv.org/abs/2108.11556.

[30] Rajaee, S., & Pilehvar, M. T. (2021). A Cluster-based Approach for Improving Isotropy in Contextual Embedding Space. http://arxiv.org/abs/2106.01183.

[31] Ren, H., & Lu, H. (2018). Compositional Coding Capsule Network with K-Means Routing for Text Classification. http://arxiv.org/abs/1810.09177.

[32] Schmidt, C. W. (2019). Improving a tf-idf weighted document vector embedding. http://arxiv.org/abs/1902.09875.

[33] Schockaert, S., & Jameel, S. (2019). Word and Document Embedding with vMF-Mixture Priors on Context Word Vectors. http://kar.kent.ac.uk/contact.html.

[34] Sinha, K., Dong, Y., Cheung, J. C. K., & Ruths, D. (2018). A Hierarchical Neural Attention-based Text Classifier (pp. 817–823). http://wiki.dbpedia.org/.

[35] Thongtan, T., & Phienthrakul, T. (2019). Sentiment Classification using Document Embeddings trained with Cosine Similarity (pp. 407–414). https://github.com/tanthongtan/dv-cosine.

[36] Wang, B. (2018). Disconnected Recurrent Neural Networks for Text Categorization (pp. 2311–2320). Association for Computational Linguistics.

[37] Werner, M., & Laber, E. (2019). Speeding up Word Mover's Distance and its variants via properties of distances between embeddings. http://arxiv.org/abs/1912.00509.

[38] Wohlwend, J., Elenberg, E. R., Altschul, S., Henry, S., & Lei, T. (2019). Metric Learning for Dynamic Text Classification. http://arxiv.org/abs/1911.01026.

[39] Wu, C., Wu, F., Qi, T., & Huang, Y. (2021). Hi-Transformer: Hierarchical Interactive Transformer for Efficient and Effective Long Document Modeling. http://arxiv.org/abs/2106.01040.

[40] Wu, L., En-Hsu Yen, I., Xu, K., Xu, F., Balakrishnan, A., Chen, P.-Y., Ravikumar, P., & Witbrock, M. J. (2018). Word Mover's Embedding: From Word2Vec to Document Embedding (pp. 4524–4534). https://github.

[41] Xiong, Y., Feng, Y., Wu, H., Kamigaito, H., & Okumura, M. (2021). Fusing Label Embedding into BERT: An Efficient Improvement for Text Classification (pp. 1743–1750).

[42] Yamada, I., & Shindo, H. (2019). Neural Attentive Bag-of-Entities Model for Text Classification. http://arxiv.org/abs/1909.01259.

[43] Yao, L., Mao, C., & Luo, Y. (2018). Graph Convolutional Networks for Text Classification. http://arxiv.org/abs/1809.05679.

[44] Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., & Ahmed, A. (2020). Big Bird: Transformers for Longer Sequences. http://arxiv.org/abs/2007.14062.

[45] Zhang, Y., Yu, X., Cui, Z., Wu, S., Wen, Z., & Wang, L. (2020). Every Document Owns Its Structure: Inductive Text Classification via Graph Neural Networks.

[46] Zhao, W., Ye, J., Yang, M., Lei, Z., Zhang, S., & Zhao, Z. (2018). Investigating Capsule Networks with Dynamic Routing for Text Classification.