A Survey of Models for Constructing Text Features to Classify Texts in Natural Language

Ksenia Lagutina, Nadezhda Lagutina
P.G. Demidov Yaroslavl State University
Yaroslavl, Russia
ksenia.lagutina@fruct.org, lagutinans@gmail.com

***Abstract*—In this survey we systematize the state-of-the-art features that are used to model texts for text classification tasks: topical and sentiment classification, authorship attribution, style detection, etc. We classify text models into three categories: standard models that use popular features, linguistic models that apply complex linguistic features, and modern universal models that combine deep neural networks with text graphs or language models. For each category we describe particular models and their adaptations, note the most effective solutions, summarize advantages, disadvantages and limitations, and make suggestions for future research.**

I. Introduction

Text classification is one of the common research topics in the field of text processing. In this area, a large number of popular tasks are investigated, such as the classification of texts into categories of domains, authorship attribution, sentiment classification, classification by genre, etc. In addition, in different domains, more specific tasks include the classification of medical texts, legal documents, fake and toxic messages, and many others. The solution to these problems includes two main parts: feature extraction and selection and classification techniques [1].

To build a text model capable of classifying a document, the basis is the extraction of numerical features [2]. The correct selection of features is the main solution to classification challenges from high dimension vectors to modeling the semantics of the domain [3]. Unfortunately, most of the works that systematize knowledge in the field of text classification are concentrated on surveys of classification methods [1], [4], [5]. Therefore, the authors of this paper set the task of considering the text classification from the point of view of modeling the numerical features of the natural language texts.

We divide approaches to text modeling into three categories. The first category (Section II) includes the most popular text features. These character and word-level features are easily computed using standard word processing tools. The second category (Section III) includes more complex features that reflect the lexical peculiarities of the language and the semantics of the domain. The third category (Section IV) include universal numerical natural language models proposed in recent years. Such models are intended to solve several different natural language processing (NLP) tasks applying the same set of text features.

II. Standard text models

A. Overview

The standard text models usually contains the text features that are very frequently used in text classification research and systematically provide good results. Nevertheless, the researchers not only apply them directly with standard classifiers, but also use methods to reduce the dimensions of feature vectors and improve performance, adapt models for specialized tasks, and give them as inputs to neural networks.

B. Standard features and standard classifiers

The main models of text representation in classification tasks are based on quite easily calculated features: bag-of-words, word embeddings, term frequency, n-grams, weighted words [1], [6]. Modern word processing libraries allow to get quickly these features for texts in different languages and even more complex ones, such as PoS-tags and the role of a word in the sentence [7]. Standard machine learning classifiers also play an important role in setting up experiments: K-Nearest Neighbor, logistic regression, Naive Bayes, Support Vector Machine, Random Forest, Neural Networks. Thus, researchers are able to quickly create a tool for the problem solution and conduct experiments.

This approach is quite successful in solving the authorship attribution problem at the PAN-2018 competition [8]. One of the PAN corpora consists of fanfiction texts written by non-professional authors. The corpus covers five languages (English, French, Italian, Polish, and Spanish). 11 teams propose the problem solution. The researchers use n-grams of characters, words, and PoS, word and sentence length, and lexical richness functions. For the authorship attribution, the support vector machine (SVM) was used (in most of the materials presented), neural networks, an ensemble of three simple models based on logistic regression, and the compression-based cosine similarity measure. The teams classify the texts by 5, 10, 15, and 20 authors. The best F-score results are 55.6 % for Polish texts and 85.6 % for Spanish. This result is provided by approaches based on n-grams of characters/words. The highest average scores are obtained for English and Spanish, while Polish texts are the most difficult to analyze. A large number of experiments allow to conclude that the number of candidate authors is inversely proportional to the accuracy of attribution, especially when more than 10

TABLE I. RESEARCH THAT PROPOSE METHODS TO SOLVE TEXT CLASSIFICATION TASKS USING STANDARD FEATURES

Authors	Classification tasks	Features	Classifier	Text corpora
Kestemont et al. [8]	Authorship attribution	Bag-of-words, word n-gram, PoS-tags, word positions	SVM, neural networks	PAN-2018
Aljumily [9]	Authorship attribution	Bag-of-words, n-grams, PoS-tags, word positions	Beta-Flexible Clustering, Hierarchical Complete Clustering, Squared Euclidean Distance	The custom corpus of Facebook texts
Shah et al. [10]	Topical classification	Bag-of-words	Logistic regression, Random Forest, KNN	BBC news
Kermani et al. [11]	Topical classification	Bag-of-words	1-nn classifier	TecTc100, Reuters-21578
Basha et al. [12]	Topical classification	Term weights	KNN, Naive Bayes	Reuters-21578, The custom corpus
Dogan and Uysal [13]	Topical classification	Bag-of-words	SVM	Reuters-21578, 20-mini-News, 20-News
Kou et al. [14]	Sentiment classification Topical classification Ad approval classification Spam classification	Bag-of-words	SVM	Pang & Lee, IMDb, Farm-ads, Spam, 20-News, Cade, Reuters-21578
Chandra [3]	Topical classification	Term Frequency, Mutual Information, Chi-square	SVM, MNB	20-News, Reuters-21578, the custom corpus
Bahassine et al. [15]	Topical classification	Bag-of-words	Decision Tree, SVM	OSAC
Pinto et al. [16]	Classification by relevance	Word n-grams, PoS-tags, LDA	Minimum Distance Classifier, SVM, Random Forest, KNN, Naive Bayes	The custom corpora of Twitter and Facebook posts and messages
Potthast et al. [17]	Style classification Fake news detection Satire detection	N-grams of characters, PoS-tags, bag-of-words	Naive Bayes	BuzzFeed-Webis Fake News, the custom corpus
Onan [18]	Style classification	Word and character n-grams, PoS-tags, bag-of-words, most discriminative words	Linear regression, Random Forest, KNN, Naive Bayes, SVM	LFA
Gargiulo et al. [19]	Multi-label classification Medical text classification	Word embeddings, word2vec, PoS-tags, Dependency Tree	Deep Neural Network	PubMed
Kim et al. [20]	Topical classification	Word embeddings	Capsule network	20-news, Reuters-10, IMDb, Movie Review
Wu et al. [21]	Topical classification Sentiment classification	Word embeddings	Siamese capsule networks	Movie Review, SUBJ, SST1, SST2
Sachan et al. [22]	Topical classification Sentiment classification	Word embeddings	Bidirectional LSTM	ACL-IMDB, Elec, AG-News, Dbpedia, RCV1, IMDB, Arxiv
Liu et al. [23]	Sentiment classification	Word embeddings	Bidirectional LSTM	Movie Review, IMDB, RT-2k, SST-1, SST-2, Subj
Liu et al. [24]	Topical classification Sentiment classification	Word embeddings	RNN	Movie Review, AG news

authors are considered, while an increase in the number of texts in the training set has a positive effect on the recognition accuracy. This conclusion is characteristic of most of the works in this area. Interestingly, the corpus of texts is selected from a specific field of fiction, written by non-professional authors.

Aljumily [9] evaluates the effectiveness of different types of text features to test their ability to be style markers for author identification in Facebook short text messages. The 328 features are divided into five types: parts of speech, function words, word bigrams, character trigrams, and token-based identifiers. To assess the impact of each group of parameters, the author uses a small hand-marked corpus of 221 texts by 10 authors and visualizes the results. The analysis shows that the best authorship attribution results are found using functional words, although the accuracy is only 60 %, the second in quality are parts of speech — 50 %. For other feature types, the attribution result is below this level: both word bigrams and symbolic trigrams reach only 40 %, and token-based identifiers provide the worst result. In addition, a fairly obvious conclusion is made that longer messages in the training set make it possible to better identify the author's style.

Although the results of this study are obtained on a very small text corpus, they show, on the one hand, the importance of choosing the text features, and on the other hand, the difficulty of interpreting the standard features.

The high quality of classification is often achieved through the selection or modification of classifiers. Shah et al. [10] has developed a classification system for BBC news texts. The authors compare logistic regression, random forest, and K-nearest neighbors as classification algorithms. The BBC news corpus is classified into 5 categories: Business, Entertainment, Politics, Sports, Technology. The vector of features for each document is formed on the basis of the bag-of-words model.

The most stable and high results are shown by the logistic regression classifier, the accuracy 97 %, F-measure 98 %. The second place is taken by a random forest classifier with an accuracy 93 %, an F-measure 95 %. The algorithm with the lowest accuracy is K-nearest neighbor with an overall accuracy of 92 %, F-measure 91 %.

Despite the high results of these classifiers, the authors point to a number of problems arising in the text classification. The same classifiers on different corpora can give very different results. The text feature vectors have a large dimension: several tens of thousands of functions, but most of them do not affect the classification result, some of them can even sharply reduce the classification accuracy. The same feature of the vector can contribute to good accuracy in one case and poor in another. A small change in the initial data can lead to a radical change in the structure of decision trees, especially when they are small in size. Raychaudhuri et al. [25] draw similar conclusions.

C. The problem of the dimension of feature vectors

The problem of the dimension of feature vectors is one of the essential problems that arises when using models based on word embeddings, term frequencies, n-grams, etc. The main methods for selecting parameters are based on mathematical methods for reducing the dimension of the feature space.

Kermani et al. [11] also point to the problem of the large dimensions of feature vectors. They use a bag-of-words model and a simple 1-nn classification algorithm to categorize news messages (4 categories from the TecTc100 corpus and 6 categories from the Reuters-21578 corpus). The article proposes a hybrid approach to selecting a subset of features. It combines feature filtering by ranking with an information retrieval method, and selection based on calculating the correlation between pairs of functions. The best quality of classification is the accuracy of 90.4 while the dimension of the feature vector is reduced from thousands to 65.

Basha et al. [12] solve the problem of classifying texts into categories. Each feature (a separate word, term, or token) is assigned a score based on the function proposed by the authors. Scoring functions include mathematical definitions and probabilistic approaches based on statistical information in documents in various categories. Terms with a higher weight are selected as elements of the feature vector. To conduct the experiments, the authors have built their own corpus of online articles from CNN, the Washington Post, and the New York Times. It includes 150 documents in the 7 categories, with an average of 702 words per document. Additionally, the corpus Reuters-21578 is used, consisting of 108 categories. The authors use the KNN classifier and the naive Bayesian classifier. They assess the precision and recall of the classification for different dimensions of the feature vectors from 250 to 2000. The quality of the solution changes greatly, including when varying the parameters of the feature scoring. Nevertheless, it turns out that the naive Bayesian classifier works better and more stable, and there is also a decrease in quality with a decrease in the dimension of the feature vectors.

Dogan and Uysal [13] propose the own scheme for weighting features for solving the problem of classifying texts into categories. They describe two new weighing schemes for the terms, derived from the standard gravity inverse moment formula, to improve the weighing behavior of the existing TF-IGM scheme. The features of the proposed schemes are compared with other term weighting schemes for both the Reuters-21578 unbalanced corpus and the balanced (20 mini-newsgroups and 20 newsgroups) corpora. Experimental results show that the proposed methods are generally superior to all standard schemes and have comparable or even better effectiveness. The F-measure is 83–95 %. The authors also compare the results of applying their method for different classifiers: KNN, SVM, Neural networks. The best is SVM. Interestingly, the F-measure decreases as the number of features increases. However, in contrast to the previous work, the dimension of the vector varies from 500 to 25000.

Kou et al. [14] compare 5 methods of feature selection at once. These methods are based on Multi Criteria Decision Making. The paper deals with the problems of binary classification sentiment analysis and multi-class text classification with small sample corpora. The stability and efficiency of the classification is assessed using 10 sets of texts. Based on the experiments, the authors make recommendations on the use of the considered methods, but conclude that none of the feature selection method provides the best performance for all criteria, regardless of the number of features and the selected classifier. The authors suggest that future studies analyze other methods and more corpora. This is an important area of research, since the described text models can be considered as universal for different subject areas and even for different languages.

Such a universal classification method is discussed by Chandra [3]. The author solves the problem of classification of news texts. It compares Term Frequency, Mutual Information, and Chi-square feature selection methods, and uses two different classifiers, SVM and Multinomial Naïve Bayes (MNB). The experiments are conducted on two commonly used English text sets: 20-Newsgroups, Reuters, and on the author's corpus of texts in Indonesian (5 categories, 9083 documents). Experiment scores range from 85 % to 90 % of the F-measure. The best result is shown by the chi-square test.

Similar results are obtained for the topical classification of Arabic texts [15]. The authors use a modified chi-square feature selection function (called ImpCHI) to improve classification efficiency. One of OSAC's open Arabic corpora is used to evaluate the quality of feature selection. The corpus contains 5070 documents of various lengths. These articles are divided into six classes. Experiments show that the combination of the ImpCHI method and the SVM classifier is superior to other combinations. The best F-measure obtained for this model is 90.50 % with 900 features.

D. Highly specialized classification tasks

The tasks considered before, as a rule, have a fairly universal formulation, which is a classification of texts by categories (topics), authors, sentiments. However, in the field of word processing, there are many problems that go beyond such definitions.

Pinto et al. [16] classify articles by relevance criteria. The authors formulate six journalistic criteria for the relevance of a piece of text. Then they classify texts according to these criteria using standard sets of text features and classifiers. The best results are obtained when relevance is predicted based on an initial prediction of each of the six accepted criteria. The accuracy and the F-measure reaches 79 % and 82 % using random forest classifiers to predict each of the criteria and a KNN classifier to predict relevance based on these intermediate predictions.

Potthast et al. [17] solve the problem of comparing the styles of fake news, satire, as well as texts written by representatives of left and right parties (hyperpartisan), and the mainstream. In fact, the task is set to classify the texts into special categories. The authors calculate the standard text features used to define the author's style, corresponding to the bag-of-words, the n-gram model, as well as domain-specific features including the ratio of cited words to external links, the number of paragraphs and their average length in the document. The experiment discards all features that are almost not represented in the documents of the corpus (i.e., they are found in less than 10 % of documents). Experiments show that hyperparty news can be well distinguished in style from the mainstream (F-measure is 78 %), as well as satire from both (F-measure is 81 %). However, the quality of fake news detection is low (F-measure is 46 %). In this work, universal characteristics and classifiers show good results only for tasks in the domain. It is hard to qualitatively classify fake news with their help. The authors draw an interesting conclusion that the writing styles of opposite orientations, namely left and right, are actually very similar: there seems to be a general style.

Besides, we would like to note the used corpus of articles. It is a large corpus of 1 627 articles that have been manually reviewed by professional BuzzFeed journalists. We can conclude that this corpus has a very high-quality markup.

In most of the studies described above, the authors use several standard classifiers and select the best results. Some researchers have achieved high quality classification of texts by modifying the classification methods. An example is the research by Onan [18], where the author classifies text documents into three classes: expressive, appellative, and informative. He proposes an ensemble schema of classifiers combined with the extraction of an efficient set of text features. The highest F-measure obtained according to the proposed scheme is 94.43 %.

E. Classification using neural networks

Methods of text classification using neural networks can be discussed separately. In this case, standard numeric features include word embedding, in particular, word2vec, and PoS-tags.

Gargiulo et al. [19] solve the multi-label text classification problem basing on the Hierarchical Label Set Expansion (HLSE) methodology used for ordering data labels. Deep neural network is proposed as the basis for solving the classification problem. To represent text, word embeddings and a module for extracting text features from the numerical representation of words using a convolutional neural network (CNN) are used. Experiments are carried out with the PubMed corpus of medical texts. Apparently due to the complexity of the task, F-measure is low — 56 %.

A capsule neural network is used to solve the problem of text classification by categories and sentiments [20]. As in the previous article, the feature vector is formed on the basis of word embeddings. For the classification of texts by the categories, the accuracy is from 86.74 % for 20 categories to 94.80 % for 6. For the sentiment classification it is from 80.98 % for the movie reviews to 90.10 % for the MPQA corpus.

A similar study of text classification by sentiments and categories using Siamese capsule networks is presented by Wu et al. [21]. The F-measure varies from 83.2 % to 96.3 %.

Sachan et al. [22] propose a bidirectional LSTM network for the task of classifying text by categories and sentiments. The basis for a successful solution is the proposed learning strategy. The text model includes standard word embeddings. The experiments are carried out with two text corpora for sentiment classification and five for topical classification. For the task of determining the positive and negative sentiments, the F-measure is 84.1 %. The authors assess the quality of the classification by error rates. Interestingly, for smaller text corpora, error rates are significantly lower: 5.62 % for AG-News (127 600 texts, 4 categories), 0.91 % for Dbpedia (630 000 texts, 14 categories), 7.78 % for RCV1 (65 385 texts, 51 categories), 35.64 % for IMDB (2,560,000 texts, 5 categories), and 31.76 % for Arxiv (1,107,00 texts, 127 categories). Additionally, the error achieves the minimum in the case when the training sample is significantly larger than the test one: 560 000 and 70 000.

Most of the works on the text classification by categories or sentiments using neural networks show high results. Therefore, such methods can be considered as the most successful universal approach in this area.

Liu et al. [23] note not only the problem of large dimension and sparseness of text data, but also the complex semantics of natural language. To solve these problems, the authors propose a new neural network architecture: bidirectional LSTM (BiLSTM), attention mechanism and the convolutional layer. The last layer extracts high-level phrase representations from word embeddings vectors, and BiLSTM is used to access both previous and subsequent context representations. The attention mechanism is used to focus information in different ways from the hidden layers of the BiLSTM. The authors claim that such an architecture can capture both the local features of phrases and the global semantics of sentences, and position their method as universal for the text classification. Experiments are conducted on six sentiment classification corpora. The accuracy for the sentiment classification reaches 84–94 %. Such high results are achieved with relatively small volumes of used text corpora from 2000 to 50 000 short messages.

It should be noted that there is lack of works that publish an expert assessment of the obtained results, and not only statistical assessments of quality. This is due, in particular, to the strong formalization and abstractness of the features of the texts in natural language, obtained using standard text processing libraries. It is also quite difficult to interpret the results of the work of classifiers for a large number of texts, especially neural networks. Expert assessment and interpretation of the results of automatic text classification, especially errors in this classification, could help improve the quality of the algorithms.

In this regard, the results of Liu et al. [24] are interesting. The authors propose Jumper, a new neural network framework that models text classification as a sequential decision-making process. The neural system sequentially scans a piece of text and makes decisions on its classification at a certain time. The basis of the system is RNN that sequentially processes incoming sentences and stores information. Based on the RNN states, the text class is predicted. This process allows to track individual steps and analyze their results, if necessary. The accuracy is 82.69 %.

F. Summary

Summing up, the standard text features of the text: bag-of-words, word embeddings, term frequencies, and n-grams model well short texts and articles for classification tasks, where the categories are quite independent from each other: positive and negative, sports and politics, etc. In the case when the classification is investigated in a narrow domain with an insufficient amount of training data, or division into categories is nontrivial, standard methods work unstable. This problem can be solved by improving the text model.

Another key to success is the quality and volume of the text corpus used to solve the problem. The creation of such corpora is a time-consuming task that receives too little attention, especially for national languages.

III. Text models with linguistic features

A. Overview

When solving text classification tasks, researchers are faced with problems associated with the limited sets of data for training, the need to take into account the peculiarities of the domain, or the specifics of the classes, the belonging to which must be determined. The researchers deal with them combining standard and complex features and applying syntactic, semantic, or more specific linguistic features.

B. Combinations of standard and complex features

Many investigators propose to solve text classification problems by using or adding more complex text features than standard ones.

Liu and Avci [26] suggest to improve classifier performance by forcing the model to focus on toxic terms. They use an L2 distance loss and task-specific prior values. These features are added to the objective function used to obtain a vector of text features, so that the use of certain keywords indicates that the text belongs to the toxic class. To mark words as toxic, an additional numerical feature (from 0 to 1) is introduced. In experiments with the corpus of texts from comments to articles from Wikipedia, the F-measure has values 70–75 %.

Khalid and Srinivasan [27] set the task of defining the style specifics of 9 online communities from 3 social media platforms discussing politics, television, and travel. The authors mainly use standard stylometric text features, such as character frequencies, parts of speech, short word counts, and add to them measures of the variety and range of the used vocabulary. In experiments to classify the belonging of a text to a particular community, it is found that style is a good indicator of group membership. The highest F-measure is 95 % for the politics community, the lowest value is 71 % for travel.

Ding et al. [28] note that determining the author's style depends on the manual selection of text features and suggest automating this process using neural networks. Stylometric features of the word level are selected taking into account Topical bias, Local contextual bias, Lexical bias. Neural networks are used to select features for each type of bias. The resulting feature vectors of texts are used in authorship characterization and authorship verification tasks. The AUROC measure achieves 0.79–0.81.

Volkova et al. [29] classify 130 thousand news posts as suspicious or verified, and predict four subtypes of suspicious news: satire, hoaxes, clickbait, and propaganda. They find out that the features reflecting the functions of social interaction are the most informative for separating the four types of news messages. These features are calculated on the basis of lexical resources containing hedges (expressions of tentativeness and possibility), assertive verbs, factive verbs, implicative verbs and report verbs, factual data, rhetorical questions, imperative commands, personal pronouns, and emotional language. As a result of using the combination of all features, the F-measure of the binary classification is 99 %, and for 4 classes it is 92 %. Such a high result is most likely due to the fact that the used lexical resources qualitatively reflect the semantic features of the considered news classes.

Škrlj et al. [30] present the tax2vec method. It find taxonomy-based features taking into account WordNet hypernyms. Then the algorithm computes TF-IDF values using word hypernyms. The number of features is reduced by the feature selection methods based on term counts, term betweenness centrality, mutual information, or PageRank-based hypernym ranking. Finally, feature vectors are classified by SVM, HILSTM, and deep feedforward neural networks. The approach is investigated for short text classification tasks: author profiling, topical classification, and biomedical text classification. The accuracy for all varies significantly from 50 % to 95 % for different algorithm parameters. But despite the instability of the method, the use of proposed features can be successfully interpreted from the linguistics point of view.

Another example of lexical resource usage is the research [31]. The authors examine the Extreme Multi-Label Text Classification challenge in the field of legislation. They annotate a corpus of 57 000 legal documents with labels that

TABLE II. Research that propose methods to solve text classification tasks using linguistic features

Authors	Classification tasks	Features	Classifier	Text corpora
Liu and Avci [26]	Toxic text classification	Word embeddings, keywords, L2 distance loss, task-specific prior values	CNN	The corpus of comments from Wikipedia
Khalid and Srinivasan [27]	Style classification	PoS frequencies, character frequencies, diversity and range of the vocabulary	Random Forest	The custom corpus of comments from 4chan, Reddit, and Voat
Ding et al. [28]	Authorship characterization Authorship verification	Syntactic modality, PoS tags, word embeddings, topical, local contextual, and lexical bias	Neural Network	PAN 2014 , ICWSM 2012 Twitter
Volkova et al. [29]	Suspicious text classification	Subjectivity, psycholinguistic, bias, and moral foundation cues, hedges, assertive, factive, implicative, report verbs, LIWC	Neural Network	The custom corpus of tweets
Škrlj et al. [30]	Author profiling Topical classification Biomedical text classification	Taxonomy-based features	SVM, HILSTM, Deep Feedforward NN	PAN, MBTI, BBC News, Drugs
Chalkidis et al. [31]	Legislation text classification	Bag-of-words, thesaurus-based features	Neural Network	EURLEX 57
Khairova et al. [32]	Style classification	Syntactic dependency relations	Random Forest	The custom corpus of articles from Wikipedia, blogs, news
Zhou et al. [33]	Satire detection	Semantic features based on inconsistencies in the sentence structure	Game-theoretic rough set decision model	The custom corpus of tweets
Horne and Adali [34]	Fake news detection	PoS, stopwords, punctuation, negations, informal/swear words, interrogatives, capital letters, syntax tree depth, readability, bag-of-words sentiment	SVM	The custom corpus of Buzzfeed and Facebook news
Vajjala [35]	Automatic essay scoring	Type-token ratio, lexical diversity, document length, PoS tags, syntactic trees, phrasal groups, errors	Logistic regression	TOEFL11, FCE
Liu et al. [36]	Biomedical text classification	List of regular expressions, n-grams	Naive Bayes, SVM, RNN, CNN	The custom corpus
Almatarneh and Gamallo [37]	Sentiment classification	N-grams, uppercase letters, intensifier, negation, elongated words, Doc2Vec, sentiment lexicons	SVM	HotelExpedia
Levitan et al. [38]	Deception detection	Pronouns, articles, formality measures, hedge words, filled pauses, laughter, complexity, contractions, denials, affect language,specificity score, NEO-FFI, follow-up questions, individual traits	Random Forest, Logistic Regression, SVM	CXD
Potha and Stamatatos [39]	Authorship verification	Bag-of-words, LSI, LDA	KNN	PAN-2014, PAN-2015
Sinoara et al. [40]	Topical classification	Word2Vec, word- and word-sense embeddings, semantic network	Naive Bayes, J48, SVM, KNN, IMBHN	CSTR, Ohsumed-400, BBC, SemEval-2015, BS-Top4
Ballier et al. [41]	Language level classification	Errors	Logistic regression, Random Forest, KNN, Naive Bayes, SVM	EFCAMDAT
Baumann et al. [42]	Style classification	Character-by-character encoding	RNN	German Text Archive
Plecháč et al. [43]	Authorship attribution	Stressed syllables, sounds, frequent words, character n-grams, word n-grams	SVM	Corpus of Czech Verse, Metricalizer, Corpus de Sonetos del Siglo de Oro, Chicago Rhyming Poetry Corpus
Balint et al. [44]	Genre classification	Word length, n-grams, PoS, commas, sentence boundaries, phonetical repetition	DFA	USE
Lagutina et al. [45], [46]	Classification by time periods Authorship verification	Average sentence length, frequencies of characters, average word length, n-grams, rhythm features	AdaBoost, Random Forest, Bidirectional LSTM, GRU	The custom corpora of literary texts

correspond to the concepts of the EUROVOC multidisciplinary thesaurus. The text vector is constructed basing on standard bag-of-words features. Multi-Label Text Classification is carried out using a neural network, taking into account the structure of the text. The F-measure is about 70 %.

C. Features based on the sentence structure

A significant part of researchers use the text features based on the structure of the sentence. Modern word processing tools can analyze this structure for different languages, highlight the roles of words in a sentence, and build a dependency tree for each sentence separately.

Such a tool is applied by Khairova et al. [32]. They use 7 dependency representation of UD evolves out of Stanford Dependencies, which follows ideas of grammatical relations-focused description. Each text is characterized by a vector of dependencies. This model of texts is used by the authors to classify texts by style. The experiments are carried out on text corpora based on Wikipedia, social, and media texts. The recall achieves 88.7 % and precision achieves 88.8 % in the case of the classification of texts by Blogs, News, and Wikipedia corpora.

Zhou et al. [33] solve the problem of detecting satirical news. They draw attention to the fact that in this area the semantic text features are almost never used. As such features, they consider inconsistencies in phrases, entities, as well as between the main and relative sentences, that they find based on the analysis of the sentence structure. Then, to detect satirical text the authors use the game-theoretic rough set decision model instead of the standard classifier. The accuracy is 82.71 %. The researchers claim that their approach works well in difficult cases where it is hard to separate satire and short news stories.

Horne and Adali [34] apply a very large set of text features at various levels to solve the problem of fake news detection. As elements of the feature vector, they use simple word-level features: PoS tags, number of informal words, stopwords, punctuation, negations, informal/swear words, interrogatives, words that consist of capital letters, sentence syntax tree depth, noun phrase syntax tree depth, and verb phrase syntax tree depth. In addition, the authors compute the readability of each text. The classification quality achieves between 71 % and 91 % of the accuracy in separating stories from real news. The use of such high-level text features allows the authors to conduct a qualitative analysis of the classification results. They disprove the assumption that fake news is written to look like the real thing. Fake news is more like satire than real news, so the authors conclude that the belief in fake news is not achieved by the power of arguments, but by the creation of certain associations between entities and statements.

Vajjala [35] solves the automatic essay scoring problem. The author also uses a variety of text features: type-token ratio, measure of textual lexical diversity, document length, PoS tags, syntactic parse trees of sentences, average number and size of various phrasal groups, and measures of parse tree height per sentence. In addition, the author evaluates text coherence by analyzing the structure of sentences. The accuracy achieves 73.2 %. The researcher also notes that little is investigated about that specific linguistic features are useful for predicting scores.

Vajjala [35], when evaluating text coherence by analyzing the structure of sentences, finds certain patterns of using words and phrases. This approach is typical for the tasks of extracting information from text. However, it is also used for classification tasks, for example, for the classification of medical texts [36]. Liu et al. propose a regular expression-based method using genetic programming to evolve regular expressions that can classify medical text query. The result accuracy becomes very different for various categories. But for most categories it is at least 85 %. The method generates classifiers that can be fully understood, verified, and updated by doctors, that is fundamental to medical practice.

D. Semantic features

The analysis of the structure of sentences allows to highlight the semantic features of the text [26]. Such features are very important for natural language processing tasks.

Almatarneh and Gamallo [37] identify extreme opinions (the most negative and the most positive) for hotel reviews. To extract word semantics of within the considered domain, the authors calculate text functions based on the use of words that enhance opinion and sentiment dictionary, simultaneously with the standard features of the word level (n-grams) and document level (Doc2Vec). The dictionary is formed basing on the analysis of tagged hotel reviews. In fact, the authors propose a method for forming a complex lexical resource of the domain in combination with the task of sentiment analysis. The results of reviews classification show the best F-measure value of 73–76 % when using all types of features.

Lexical resources such as dictionaries and thesauri are one of the simplest ways to describe the semantics of a domain. However, this approach is very time consuming, since for many tasks these resources either do not exist or cannot be used directly in software systems.

To detect deception in interview dialogues, Levitan et al. [38] use both the Dictionary of Affect Language to measure the emotional meaning of texts and the more complex proprietary resource Linguistic Inquiry and Word Count (LIWC). LIWC is a corpus and text analysis program that counts words in psychologically meaningful categories. With the help of these resources, the main features of the text are formed. The authors add the parameters of the respondent's profile to the feature vector: gender, native speaker or non-native speaker. The combination of linguistic and individual feature make it possible to obtain the best F-measure of 72.74 %.

To determine semantic relationships and dependencies, researchers sometimes use the Latent semantic analysis, LSA and its varieties, Latent Semantic Indexing (LSI), Latent Dirichlet allocation (LDA). These methods are used to isolate key domain concepts and identify related words.

The usefulness of LSA and LDA topic modeling methods is investigated in authorship verification by Potha and Stamatatos [39]. The researchers note that these methods allow to detect hidden structures in texts, that can be considered as hidden semantic structures. The extracted structures can capture the style information of the author. The proposed approach does not depend on the genre of the text and is very promising for further research. Average AUC is 0.86.

Text features based on the LSA method are included in the parameter set for evaluating the quality of essays [47]. The purpose of the study is to investigate change in linguistic constructs over time. Indexes obtained in the LSA expand the set of references to semantically related words and allow to determine the coherence of the text.

LSA methods are most effective with large corpora of domain texts. Sinoara et al. [40] propose two approaches to the semantic representation of document collections, NASARI + Babel2Vec and Babel2Vec, based on word sense disambiguation and embeddings of words and word senses. The resulting feature vectors do not require a large number of texts for training of classification models. The proposed approaches are applied to the multi-class classification problem. The number of classes in text collections used in the experiments was from 3 to 23. The F-measure is very high in the classification with a small number of classes (99 %, 3 or 4 classes) and very low for a large number (40 %, 23 classes). The authors note the independence of the methods from the language and the volume of the training sample, although they use open lexical resources such as WordNet and Wikipedia.

E. New feature types

In several works, researchers offer new non-standard features of the text. As a rule, they rely on the features used in the studies of classical linguistics.

Ballier et al. [41] construct a classification model of learner language levels. The authors use 24 error types such as punctuation, spelling, morphosyntactic errors, syntax, lexical or collocation errors. These features are used to automatically classify language levels in the Common European Framework of Reference. The experiments are carried out on the basis of manually annotated errors in the student texts. The accuracy achieves 70 %. In addition to the classification task, the authors conduct a study of what types of errors are key features in determining language levels.

For the analysis of poetry, researchers use rhythm features. Most often there are phonetic ones based on the alternation and repetition of sounds. Baumann et al. [42] classify poems by style. They define the features of the poetic text at the character level without dividing the text into words. A recurrent neural network is used to analyze feature vectors. The F-measure of the classification into six styles is 73 %.

Plecháč et al. [43] use rhythm features to determine the authorship of poetic texts. The phonetic aspects of the text rhythm are used as features: the stress profile and the frequencies of particular sounds. The authors have collected four corpora of poetic texts: Czech, German, Spanish, and English. They experiment with comparing phonetic features with other feature sets of frequent words, character n-grams, word n-grams. The best result is obtained with a combination of all types of feature. The accuracy is from 84.5 % to 99.3 %.

Balint et al. [44] classify texts by style. The authors investigate the use of rhythm features for the classification of prose texts. The basis of these features is phonetical repetition: assonance, alliteration, rhyme. The authors note that using only eight rhythm features, documents can be successfully assigned to a specific genre with an accuracy of 81.51 %.

The authors of this review, with the participation of linguistic experts, have identified the rhythm features of the level of words and sentences (anaphora, aposiopesis, etc.). We have developed methods for constructing rhythm feature vectors of text and applied them to the analysis of literary texts. In the work [45] the problem of text classification by time periods is solved. In [46], these parameters are found to be effective in the authorship verification. Rhythm features are used in combinations with character and word-level features. A comparative study of the influence of different types of features on the quality of classification shows that rhythm features can be independent markers of style. The quality of the authorship verification is quite high and achieves up to 86–97 %.

F. Summary

As we can seen from this part of the survey, complex linguistic features are used in solving a wide range of problems in computational linguistics. Quite often, they are used in cases where it is required to identify the specifics of the text style, for example, to determine authorship or genre. Many of these features are difficult to calculate. However, a number of them have become so popular that the corresponding tools (PoS-tags, dependency trees) are added to word processing tools like NLTK, Stanza, etc. Nevertheless, the influence of different feature types on the quality of text classification tasks has not been studied enough. Systematization and identification of general trends in this area is a very difficult but important task.

To carry out such fundamental studies, large, well-labeled data corpora are required. Unfortunately, the construction of such corpora is a serious problem, as it requires significant efforts. This process is currently poorly automated, especially for national languages [14], [48]. It is clearly seen from the reviewed studies that researchers very often create their own text corpora, and often these corpora are small in size. The creation of corpora that are marked up for solving various problems, is also important for improving the performance of supervised classifiers.

Separately, we note the approaches to the construction of features that reflect the semantics of the domain. Most often they are based on linguistic resources such as dictionaries, encyclopedias, and thesauri. There are few such resources available today, primarily for the English language. Unfortunately, for many domains and specific tasks, these resources are suitable with great limitations. Thus, methods of automating the construction of linguistic resources for different languages are a promising area of work.

IV. Modern universal text models

A. Overview

In recent years researchers often propose text models based on deep neural networks. They can be applied to different text classification tasks without additional dependencies from the text type and domain. The most popular models are features based on graph networks, and word embeddings pretrained on the large corpus.

B. Graph-based approaches

The graph-based approaches construct graphs for particular texts or aggregate them for several documents, and process them by special networks.

The TextING [49] tool classifies texts building individual graphs for each document and applying Graph Neural Networks (GNN). The authors model texts basing on their keywords and relationships between them. The graph of a text contains unique words as vertices and their occurrences as edges. Then the graph nodes are represented as embeddings and given as inputs to the GNN. The quality of the method is measured for two NLP tasks: sentiment and topical classification. The accuracy for the sentiment classification of movie reviews achieves 80 %, topical classification of news — 98–95 %, medical abstracts — 70 %. So this approach suits better for the classification of non-specific texts on different topics.

The framework TensorGCN [50] builds graph tensors for individual texts. The authors propose three tensor types: semantic, syntactic, and sequential-based features. The semantic features are captured by LSTM and filtered with the threshold for cosine similarity. The syntactic features describe dependencies between words. The sequential features are based on local co-occurrences of words. The classifier for tensors is Graph Convolutional Networks (GCN). The NLP tasks and the text corpora are the same as in the previous research adding only the 20Newsgroups corpus. The sentiment classification accuracy is about 78 %, topical classification accuracy is from 70 to 98 %. The results are very close to the TextING [49] tool.

Hu et al. [51] solve the task of short text classification. They propose a heterogeneous information network (HIN) for text modeling and Heterogeneous Graph ATtention networks (HGAT) that embed the HIN based on a dual-level attention mechanism. The text model for HIN includes standard text features: latent topics (LDA) and entity embeddings (word2vec) based on the Wikipedia corpus. The result of the HIN is presented for HGAT as the graph with documents, topics, and entities as nodes. The document node corresponds to the TF-IDF vector, the topic node — the word distribution used to represent the topic, the entity node — concatenation of its embedding and TF-IDF vector of its Wikipedia description. The HGAT method is tested on the classification by sentiments and topics. The accuracy achieves quite low values: 62 % for sentiment classification, 42 % for medical texts, about 62–72 % for news, and 82 % for snippets.

Huang et al. [52] build a text graph with word embeddings as nodes and relations with adjacent words as edges. Such text model is processed by GNN that applies message passing mechanism for convolution. The classification by topics shows 70 % of accuracy for medical texts, and 94–97 % — for news.

C. Approaches with pre-training

The more popular text models requires additional pre-training on the large text corpus before usual phases of text classification: training and testing. The corpus for pre-training can contains only raw texts without additional marks and labels. During pre-training the algorithm builds word embeddings basing on the word context. The set of such embeddings forms the language model. Then the algorithm fine-tunes the model for the particular classification task instead of the training and validate or test the classifier. In most cases pre-training, fine-tuning and testing is performed with transformer neural networks.

The ELMo [53] language model generates word embeddings applying the bidirectional LSTM neural network to word context, i.e., tokens that appear before the word. It should be pre-trained on the large unlabeled text corpora, so we get numerical vector representations of particular words. Then these embeddings can be used for specific NLP tasks. Experiments shows good accuracy and F-measure over 85–90 % for textual entailment and named entity recognition. But when the authors classify text by sentiments, the accuracy is low, only 54 %. So we can conclude that this model needs additional modifications to suit better to text classification tasks.

The GPT and GPT-2 [54], [55] language models creates word embeddings basing on the word context window of the fixed size. The neural network for training and testing is the multi-layer Transformer decoder. Like the previous model, it is also evaluated for different NLP tasks, including the text classification by sentiments and by sentence grammar. The accuracy of 91 % for sentiment classification is very high, but for sentence grammar is low — 45 %. The authors note that the larger text corpora on the pre-training phase lead to the better results.

The BERT [56] language model also uses word context to generate word embeddings, but it takes both left and right contexts. It complicates the pre-training but allows to simplify fine-tuning. The used neural network is the multi-layer bidirectional Transformer encoder. The authors compare model performance on different classification tasks including two text classification ones. The result outperforms other language models: the accuracy of sentiment classification is 95 % and of the classification by sentence grammar — 60 %. Such quality proves that the methods based on the transformer neural networks are some of the best in the state-of-the-art in text classification.

Guo et al. [57] propose to take into account positions of words in texts. They create word positional embeddings

TABLE III. Research that propose methods to solve text classification tasks using modern neural networks

Authors	Classification tasks	Features	Classifier	Text corpora
Zhang et al. [49]	Sentiment classification Topical classification	Document graphs based on keywords Graph-based embeddings	GNN	Movie Review, R8 and R52, Ohsumed
Liu et al. [50]	Sentiment classification Topical classification	Document graphs based on semantic, syntactic, and sequential features	GCN	20-News, Movie Review, R8 and R52, Ohsumed
Hu et al. [51]	Sentiment classification Topical classification	Document graphs based on TF-IDF, LDA, and word2vec entity embeddings	HGAT (based on GCN)	AGNews, Snippets, Ohsumed, TagMyNews, Movie Review, Twitter
Huang et al. [52]	Topical classification	Document graphs based on word embeddings	GNN	R8 and R52, Ohsumed
Peters et al. [53]	Sentiment classification	LSTM context-based word embeddings — ELMo	LSTM	SST-5
Radford et al. [54], [55]	Sentiment classification Grammar classification	Word embeddings based on the context window — GPT	Transformer decoder	SST-2, CoLA
Devlin et al. [56]	Sentiment classification Grammar classification	Word embeddings based on the left and right context — BERT	Bidirectional Transformer encoder — BERT	SST-2, CoLA
Guo et al. [57]	Sentiment classification	Word embeddings based on the word position	Multi-scale Transformer	SST, MTL-16
Mekala and Shang [58]	Topical classification	BERT pretrained on the contextualized corpus	HAN	NYT, 20-News
Takayama et al. [59]	Sentiment classification Topical classification	BERT and CBOW embeddings	BERT, HAN	CR, SST-5, SUBJ, arXiv
Lu et al. [60]	Sentiment classification Grammar classification Hate speech detection	GCN graph embeddings, BERT word embeddings	BERT	Movie Review, SST-2, CoLA, ArangoHate, FountaHate

applying the multi-scale transformer neural network. Besides, the authors add to text model character-level features: capitalization features, lexicon features, etc. The quality of sentiment analysis achieves 80–90 %.

Several authors modify pre-training of the BERT model.

Mekala and Shang [58] describe the ConWea framework that leverages contextualized representations of word occurrences and seed word information. They create the contextualized text corpus and apply BERT for pre-training and Hierarchical Attention Networks (HAN) for classification by topic. The best classification quality reaches 83–96 % of the F-measure.

Takayama et al. [59] propose the multi-tasking learning model that take into account negative examples, i.e. texts with different labels. It is based on BERT and CBOW embeddings. This approach is successfully applied to sentence and text classification tasks. The accuracy for sentiment classification achieves 86–96 %, for topical classification — only 36 %.

The BERT-based approach can be united with the graph-based method. Lu et al. [60] combine graph embeddings get from GCN, and word embeddings get from BERT, and apply BERT as a classifier. This approach allows to take into account both local text features based on context, and global ones computed using the graph. The authors experiment with sentiment classification, sentence grammar classification, and hate speech detection tasks. The best results in every case are over 80 % of the F-measure.

All proposed approaches are summarized in the Table III. We can see that the new universal models are tested on the narrow set of classification tasks and the limited set of text corpora.

Several researchers compare these models for specific classification tasks.

Mascio et al. [61] compare different embedding-based text models for classification of electronic health records. Although the BERT and BioBERT (BERT, pre-trained on biomedical texts) models show 91–98 % of the F-measure, the Bi-LSTM neural network with word2vec embeddings based on the MIMIC dataset, slightly outperforms transformer-based models by 1–3 %. Thus, word embeddings based on context and combined with neural networks seem prominent for text classification of specific text corpora.

Chalkidis et al. [62] solve the task of large-scale multi-label text classification (LMTC) in the legal domain. They compare several attention-based neural networks that create language models using the word context, including HAN, BERT, ELMo, etc. The BERT model achieves the best results over than 70–80 % of the precision and F-measure for different numbers of labels. The authors admit that the chosen methods are unsuitable for multi-label text classification with hundreds of thousands of labels, because of high computational costs and the fact that the larger the number of labels means the worse the classification quality, especially for non-frequent labels.

D. Summary

Thus, graph-based methods are good for topical classification of news, but do not provide very high results for sentiment classification and classification of texts from specific domains like medicine. Most probably, the proposed text models can be extended with more deep linguistic features to better classify specific texts.

Among the approaches based on pre-trained language models, the BERT model shows the best results in the state-of-the-art in text classification. Therefore, it seems to be the most promising for modification and additional adjustment for a specific task or text domain.

Nevertheless, the new language models based on transformer neural networks remain under-researched in the field of text classification tasks. They shows the great results for topical classification of news and sentiment classification of reviews. For more specific text domains like medicine they require modifications at least in pre-training. So, comparison and combination of these language models with linguistic features seems to be a promising direction of future research.

V. Conclusion

We consider various approaches to modeling text in natural language for classification tasks. The main problems of standard models are the large dimension of feature vectors and a limited range of classification tasks. The linguistic feature are very diverse, allow taking into account the specifics of tasks in domains. However, they either require the development of algorithms with complex feature calculations, or are based on voluminous linguistic resources. Modern universal models are good for solving standard tasks: sentiment and topical classification of texts. However, to cope with more specific problems, these models also require adaptation. A common problem is the construction of high-quality text corpora for machine learning and experiments that can be published in open access.

The most interesting and promising for further research are linguistic features and universal language models. On the one hand, they have a great potential for application to solving specialized tasks. On the other hand, further systematization and analysis of the results of their use in text processing tasks is required.

Acknowledgment

Ksenia Lagutina is supported by the President of Russian Federation Scholarship for young scientists and postgraduates No. SP-2109.2021.5.

References

[1] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text classification algorithms: A survey,” *Information*, vol. 10, no. 4, pp. 150 (1–68), 2019.

[2] H. Liang, X. Sun, Y. Sun, and Y. Gao, “Text feature extraction based on deep learning: a review,” *EURASIP journal on wireless communications and networking*, vol. 2017, no. 1, pp. 1–12, 2017.

[3] A. Chandra, “Comparison of feature selection for imbalance text datasets,” in *2019 International Conference on Information Management and Technology (ICIMTech)*, vol. 1. IEEE, 2019, pp. 68–72.

[4] A. I. Kadhim, “Survey on supervised machine learning techniques for automatic text classification,” *Artificial Intelligence Review*, vol. 52, no. 1, pp. 273–292, 2019.

[5] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, “Comparing automated text classification methods,” *International Journal of Research in Marketing*, vol. 36, no. 1, pp. 20–38, 2019.

[6] K. Lagutina, N. Lagutina, E. Boychuk, I. Vorontsova, E. Shliakhtina, O. Belyaeva, I. Paramonov, and P. Demidov, “A survey on stylometric text features,” in *Proceedings of the 25th Conference of Open Innovations Association (FRUCT)*. IEEE, 2019, pp. 184–195.

[7] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, “Stanza: A python natural language processing toolkit for many human languages,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020, pp. 101–108.

[8] M. Kestemont, M. Tschuggnall, E. Stamatatos, W. Daelemans, G. Specht, B. Stein, and M. Potthast, “Overview of the author identification task at pan-2018: cross-domain authorship attribution and style change detection,” in *Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018/Cappellato, Linda [edit.]; et al.*, 2018, pp. 1–25.

[9] R. Aljumily, “Evaluation of the performance and efficiency of the automated linguistic features for author identification in short text messages using different variable selection techniques,” *Studies in Media and Communication*, vol. 6, no. 2, pp. 83–102, 2018.

[10] K. Shah, H. Patel, D. Sanghvi, and M. Shah, “A comparative analysis of logistic regression, random forest and knn models for the text classification,” *Augmented Human Research*, vol. 5, no. 1, pp. 1–16, 2020.

[11] F. Z. Kermani, E. Eslami, and F. Sadeghi, “Global filter–wrapper method based on class-dependent correlation for text classification,” *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 619–633, 2019.

[12] S. R. Basha and J. K. Rani, “A comparative approach of dimensionality reduction techniques in text classification,” *Engineering, Technology & Applied Science Research*, vol. 9, no. 6, pp. 4974–4979, 2019.

[13] T. Dogan and A. K. Uysal, “Improved inverse gravity moment term weighting for text classification,” *Expert Systems with Applications*, vol. 130, pp. 45–59, 2019.

[14] G. Kou, P. Yang, Y. Peng, F. Xiao, Y. Chen, and F. E. Alsaadi, “Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods,” *Applied Soft Computing*, vol. 86, p. 105836, 2020.

[15] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, “Feature selection using an improved chi-square for arabic text classification,” *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 2, pp. 225–231, 2020.

[16] A. Pinto, H. G. Oliveira, Á. Figueira, and A. O. Alves, “Predicting the relevance of social media posts based on linguistic features and journalistic criteria,” *New Generation Computing*, vol. 35, no. 4, pp. 451–472, 2017.

[17] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, “A stylometric inquiry into hyperpartisan and fake news,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 231–240.

[18] A. Onan, “An ensemble scheme based on language function analysis and feature engineering for text genre classification,” *Journal of Information Science*, vol. 44, no. 1, pp. 28–47, 2018.

[19] F. Gargiulo, S. Silvestri, M. Ciampi, and G. De Pietro, “Deep neural network for hierarchical extreme multi-label text classification,” *Applied Soft Computing*, vol. 79, pp. 125–138, 2019.

[20] J. Kim, S. Jang, E. Park, and S. Choi, “Text classification using capsules,” *Neurocomputing*, vol. 376, pp. 214–221, 2020.

[21] Y. Wu, J. Li, J. Wu, and J. Chang, “Siamese capsule networks with global and local features for text classification,” *Neurocomputing*, vol. 390, pp. 88–98, 2020.

[22] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, “Revisiting lstm networks for semi-supervised text classification via mixed objective function,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6940–6948.

[23] G. Liu and J. Guo, “Bidirectional lstm with attention mechanism and convolutional layer for text classification,” *Neurocomputing*, vol. 337, pp. 325–338, 2019.

[24] X. Liu, L. Mou, H. Cui, Z. Lu, and S. Song, “Finding decision jumps in text classification,” *Neurocomputing*, vol. 371, pp. 177–187, 2020.

[25] K. Raychaudhuri, M. Kumar, and S. Bhanu, “A comparative study and performance analysis of classification techniques: Support vector machine, neural networks and decision trees,” in *International Conference on Advances in Computing and Data Sciences*. Springer, 2016, pp. 13–21.

[26] F. Liu and B. Avci, “Incorporating priors with feature attribution on text classification,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 6274–6283.

[27] O. Khalid and P. Srinivasan, “Style matters! investigating linguistic style in online communities,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, 2020, pp. 360–369.

[28] S. H. Ding, B. C. Fung, F. Iqbal, and W. K. Cheung, “Learning stylometric representations for authorship analysis,” *IEEE transactions on cybernetics*, vol. 49, no. 1, pp. 107–121, 2017.

[29] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas, "Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 647–653.

[30] B. Škrlj, M. Martinc, J. Kralj, N. Lavrač, and S. Pollak, "tax2vec: Constructing interpretable features from taxonomies for short text classification," *Computer Speech & Language*, vol. 65, p. 101104, 2021.

[31] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Extreme multi-label legal text classification: A case study in eu legislation," in *Proceedings of the Natural Legal Language Processing Workshop*, 2019, pp. 78–87.

[32] N. Khairova, W. Lewoniewski, K. Wecel, M. Orken, and M. Kuralai, "Comparative analysis of the informativeness and encyclopedic style of the popular web information sources," in *International Conference on Business Information Systems*. Springer, 2018, pp. 333–344.

[33] Y. Zhou, Y. Zhang, and J. Yao, "Satirical news detection with semantic feature extraction and game-theoretic rough sets," in *International Symposium on Methodologies for Intelligent Systems*. Springer, 2020, pp. 123–135.

[34] B. Horne and S. Adali, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, no. 1, 2017, pp. 759–766.

[35] S. Vajjala, "Automated assessment of non-native learner essays: Investigating the role of linguistic features," *International Journal of Artificial Intelligence in Education*, vol. 28, no. 1, pp. 79–105, 2018.

[36] J. Liu, R. Bai, Z. Lu, P. Ge, U. Aickelin, and D. Liu, "Data-driven regular expressions evolution for medical text classification using genetic programming," in *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.

[37] S. Almatarneh and P. Gamallo, "Linguistic features to identify extreme opinions: an empirical study," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2018, pp. 215–223.

[38] S. I. Levitan, A. Maredia, and J. Hirschberg, "Linguistic cues to deception and perceived deception in interview dialogues," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1941–1950.

[39] N. Potha and E. Stamatatos, "Improving author verification based on topic modeling," *Journal of the Association for Information Science and Technology*, vol. 70, no. 10, pp. 1074–1088, 2019.

[40] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, and S. O. Rezende, "Knowledge-enhanced document embeddings for text classification," *Knowledge-Based Systems*, vol. 163, pp. 955–971, 2019.

[41] N. Ballier, T. Gaillat, A. Simpkin, B. Stearns, M. Bouyé, and M. Zarrouk, "A supervised learning model for the automatic assessment of language levels based on learner errors," in *European Conference on Technology Enhanced Learning*. Springer, 2019, pp. 308–320.

[42] T. Baumann, H. Hussein, and B. Meyer-Sickendiek, "Style detection for free verse poetry from text and speech," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1929–1940.

[43] P. Plecháč, K. Bobenhausen, and B. Hammerich, "Versification and authorship attribution. a pilot study on czech, german, spanish, and english poetry," *Studia Metrica et Poetica*, vol. 5, no. 2, pp. 29–54, 2018.

[44] M. Balint, M. Dascalu, and S. Trausan-Matu, "Classifying written texts through rhythmic features," in *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*. Springer, 2016, pp. 121–129.

[47] C. A. MacArthur, A. Jennings, and Z. A. Philippakos, "Which linguistic features predict quality of argumentative writing for college basic writers, and how do those features change with instruction?" *Reading and Writing*, vol. 32, no. 6, pp. 1553–1574, 2019.

[45] K. Lagutina, N. Lagutina, E. Boychuk, and I. Paramonov, "The influence of different stylometric features on the classification of prose by centuries," in *Proceedings of the 27th Conference of Open Innovations Association FRUCT*. IEEE, 2020, pp. 108–115.

[46] K. Lagutina, N. Lagutina, E. Boychuk, V. Larionov, and I. Paramonov, "Authorship verification of literary texts with rhythm features," in *Proceedings of the 28th Conference of Open Innovations Association FRUCT*. IEEE, 2021, pp. 240–251.

[48] V. Marivate and T. Sefara, "Improving short text classification through global augmentation methods," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2020, pp. 385–399.

[49] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, and L. Wang, "Every document owns its structure: Inductive text classification via graph neural networks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 334–339.

[50] X. Liu, X. You, X. Zhang, J. Wu, and P. Lv, "Tensor graph convolutional networks for text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 8409–8416.

[51] H. Linmei, T. Yang, C. Shi, H. Ji, and X. Li, "Heterogeneous graph attention networks for semi-supervised short text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 4823–4832.

[52] L. Huang, D. Ma, S. Li, X. Zhang, and W. Houfeng, "Text level graph neural network for text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3435–3441.

[53] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 2227–2237.

[54] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[55] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, pp. 9–33, 2019.

[56] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[57] Q. Guo, X. Qiu, P. Liu, X. Xue, and Z. Zhang, "Multi-scale self-attention for text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 7847–7854.

[58] D. Mekala and J. Shang, "Contextualized weak supervision for text classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 323–333.

[59] S. Ohashi, J. Takayama, T. Kajiwara, C. Chu, and Y. Arase, "Text classification with negative supervision," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 351–357.

[60] Z. Lu, P. Du, and J.-Y. Nie, "VGCN-BERT: augmenting BERT with graph embedding for text classification," in *European Conference on Information Retrieval*. Springer, 2020, pp. 369–382.

[61] A. Mascio, Z. Kraljevic, D. Bean, R. Dobson, R. Stewart, R. Bendayan, and A. Roberts, "Comparative analysis of text classification approaches in electronic health records," in *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, 2020, pp. 86–94.

[62] I. Chalkidis, E. Fergadiotis, P. Malakasiotis, and I. Androutsopoulos, "Large-scale multi-label text classification on eu legislation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 6314–6322.