MEDINFO 2021: One World, One Health – Global Partnership for Digital Innovation
P. Otero et al. (Eds.)

doi:10.3233/SHTI220075

The Classification of Scientific Abstracts Using Text Statistical Features

Timur Ishankulov[a], Gleb Danilov[a], Konstantin Kotik[a], Yuriy Orlov[b], Mikhail Shifrin[a], Alexander Potapov[a]

[a] *Laboratory of Biomedical Informatics and Artificial Intelligence, National Medical Research Center for Neurosurgery named after N.N. Burdenko, Moscow, Russian Federation*
[b] *Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, Moscow, Russian Federation*

Abstract

Automated abstracts classification could significantly facilitate scientific literature screening. The classification of short texts could be based on their statistical properties. This research aimed to evaluate the quality of short medical abstracts classification primarily based on text statistical features. Twelve experiments with machine learning models over the sets of text features were performed on a dataset of 671 article abstracts. Each experiment was repeated 300 times to estimate the classification quality, ending up with 3600 tests total. We achieved the best result (F1 = 0.775) using a random forest machine learning model with keywords and three-dimensional Word2Vec embeddings. The classification of scientific abstracts might be implemented using straightforward and computationally inexpensive methods presented in this paper. The approach we described is expected to facilitate literature selection by researchers.

Keywords:

Natural Language Processing, Neurosurgery, Machine Learning.

Introduction

Automated text classification methods could significantly facilitate scientific literature appraisal by researchers, especially in systematic reviews [1–3]. It might be valuable to save a substantial amount of time while screening articles related to the subject of interest or enable automatic literature tracking for certain topics. Health professionals could produce systematic reviews faster, increasing publication activity and sparing time to focus on extra research.

The typical process of identifying eligible studies begins with abstract screening. The scientific abstracts selection could be approached as a binary classification task. Thus, the collection of short texts is divided into two classes: relevant (e.g., related to the topic) and irrelevant. Earlier our group has conducted a study on short abstracts classification with word embeddings and shallow machine learning (ML), reaching the F1-score = 0.78 [4]. We continued with new text classification approaches utilizing pre-trained large neural language models that provided impressive gains in many natural language processing (NLP) tasks, such as text and speech processing, lexical semantics analysis, relational semantics extraction, and parsing. The abstracts classification quality was substantially improved using Bidirectional Encoder Representations from Transformers (BERT) technology to the best F1-score = 0.857 [5]. Simultaneously in that study, we evaluated the classification quality using the ensemble of shallow ML models with BERT and obtained a lower F1-score = 0.853 [5]. Therefore, we wondered if any alternative methods could augment BERT in the ensemble and improve the classification quality.

Among various approaches to improve automatic text classification, the calculation of statistical text features could be considered. F.A. Sheikha and D. Inkpen (2010) showed F1-score = 0.985 on the binary classification task of full-text documents based on the text statistical features [6]. The collection of documents contained articles from Reuters corpus, technical texts, personal letters and emails, spoken language texts. Documents were split into two classes: formal and informal texts.

Z. Faguo et al. (2010) reported that short text classification is possible using the text statistics and explicit rules and achieved the precision = 0.893, recall = 0.658 as the best result (F1-score = 0.758) [7].

This research aimed to test the quality of short medical abstracts classification primarily based on text statistical features. If the approach proves efficiency, we might consider text statistics for inclusion in ensembles with machine learning models to improve the short text classification quality.

Methods

Dataset

The dataset for experiments was obtained from PubMed while performing a systematic review of artificial intelligence (AI) applications in neurosurgery in July 2019 [8,9]. The articles were manually split into relevant and irrelevant classes. Relevant papers were eligible for inclusion in the systematic review.

Feature engineering

The basic text statistical features we considered could be divided into four groups:

- Character statistics
- Word statistics
- Statistical ratios
- Keyword statistics

At the first stage of our experiments, we calculated the set of text statistical features without keyword statistics for each document in the training subset and used only these features to train ML models.

At the second stage of our approach, we added the keyword statistics to the basic features.

At the third stage, we added three-dimensional Word2Vec vectors to the above-mentioned features. We split Word2Vec vector values into three columns and handled them as separate features of the subset.

Character statistics

Character statistics were the number of symbols belonging to a specific distinct set. Such features included:

- The number of any symbols in the document (*text_len*)
- The number of symbols mapped in the American standard code for information interchange (ASCII) table (*ascii_letters*)
- The number of uppercase letters (*uppercase*)
- The number of lowercase letters (*lowercase*)
- The number of digit symbols (*digits*)
- The number of punctuation symbols (*punctuation*). These included: dots, commas, exclamation and question marks, different parentheses, and other punctuation marks.
- The number of space characters (*spaces*). These included all the ASCII characters denoting the whitespace: the characters space, tab, linefeed, return, formfeed, and vertical tab.
- The number of special symbols (*spec_chars*) included all the symbols that did not meet in *ascii_letters*, *digits*, *punctuation*, or *space* characters.
- The number of characters used within the study results (*result_chars*): the equals sign, the colon character, the percent sign, the tilde sign.

All the character statistics were transformed using a decimal logarithm to reduce the variability in ranges.

Word statistics

Word statistics were various statistics calculated on a higher level of document entity — words and sentences. These features were:

- The number of words in the document (*n_words*). We applied a lemmatization on the entire document before counting words. This group contained only nouns, adjectives, verbs, and adverbs.
- The number of parts of speech, including the number of nouns (*n_nouns*), the number of adjectives (*n_adjectives*), the number of verbs (*n_verbs*), and the number of adverbs (*n_adverbs*) in the document.
- The number of sentences in the document (*n_sentences*).
- The number of stop-words in the document (*n_stopwords*). These words were commonly used pronouns, prepositions, conjunctions.
- The number of unique words within a document (*n_different_words*). This number was decreased after lemmatization, so *n_different_words* showed the actual quantity of unique lemmas.
- The numbers of short (*n_short_words*) and long (*n_long_words*) words. We considered the word as short if its length was less or equal to five characters and long if it was longer than five characters.
- The average word length (*mean_word_len*) in characters.
- The average sentence length (*mean_sentence_len*) in words.
- The number of abbreviations in the document (*n_abbreviations*). This text statistic was calculated before the lemmatization to keep the original abbreviations.
- The number of unique abbreviations (*n_different_abbreviations*).

All the word statistics, except for *mean_word_len* and *mean_sentence_len*, were transformed using decimal logarithm to reduce their values and the range. The values of *mean_word_len* and *mean_sentence_len* were transformed with MinMaxScaler provided by the *sklearn* python package. As a result of the transformation, the new values varied in the range between 0 and 1, where 0 was assigned to a minimal value across the training subset, and 1 was assigned to a maximum value across the training subset.

Statistical ratios

Statistical ratios reflected the ratios between various text statistics. We selected the list of ratios that qualified for the classification task:

- The ratio of *uppercase* to *text_len* (*uppercase_to_text_len*)
- The ratio of *digits* to *text_len* (*digits_to_text_len*)
- The ratio of *spec_chars* to *text_len* (*spec_chars_to_text_len*)
- The ratio of *result_chars* to *text_len* (*result_chars_to_text_len*)
- The ratio of *punctuation* to *text_len* (*punctuation_to_text_len*)
- The ratio of *n_short_words* to *n_long_words* (*n_short_words_to_n_long_words*)
- The ratio of *n_nouns* to *n_words* (*n_nouns_to_n_words*)
- The ratio of *n_adjectives* to *n_words* (*n_adjectives_to_n_words*)
- The ratio of *n_verbs* to *n_words* (*n_verbs_to_n_words*)
- The ratio of *n_adverbs* to *n_words* (*n_adverbs_to_n_words*)
- The squared ratio of *mean_word_len* to *n_words* (*mean_word_len_to_n_words_squared*). This ratio was calculated by finding the ratio of *mean_word_len* to *n_words*, then adding 1 to the ratio and square it to expand the range of values.
- The squared ratio of *n_abbreviations* to *n_words* (*n_abbreviations_to_n_words_squared*). This ratio was calculated in the same way as *mean_word_len_to_n_words_squared*: first finding the ratio of *n_abbreviations* to *n_words*, then adding 1

to the ratio and square it to expand the range of values.

We did not apply any additional transformations (scaling or logarithmic) to the statistical ratios.

Keyword statistics

Keyword statistics represented the occurrence of the pre-defined keywords. Having expertise in the subject domain, we were able to add some keywords as features to the ML models. Keyword statistics had the potential to improve the classification quality. However, it is possible to run our algorithms without keyword statistics in case they are not available.

Initially we defined the set of keywords we wanted to track: machine, learning, learn, training, train, algorithm, model, accuracy, sensitivity, specificity, score, predict, predictive, feature. Lemmatization preprocessing depressed the morphological diversity, but some terms, such as "training" and "train", remained in adjective and noun forms of part of speech accordingly. Thus, we consolidated similar keywords into comprehensive terms.

The final list of keywords: machine, learn, train, algorithm, model, accuracy, predict, feature was labeled as *keyword_machine*, *keyword_learn*, *keyword_train*, *keyword_algorithm*, *keyword_model*, *keyword_accuracy*, *keyword_predict*, *keyword_feature* accordingly.

We calculated the number of each keyword occurrence in the documents. An overall counter of all keywords (*n_keywords*) was set as a separate feature.

Word2Vec embeddings

We proposed adding three-dimensional (the dimensionality was established experimentally) Word2Vec (W2V) vectors as features to improve the classification quality. W2V embeddings were calculated on the training subset. Three new columns, corresponding to the three dimensions of each document vector, were added to the train and test subsets: *w2v_0, w2v_1, w2v_2*.

Dataset characteristics

Major dataset characteristics are shown in Table 1.

Table 1 – Dataset characteristics

Characteristic	Min	Max	Mean
text_len	300	3261	1610.87
n_words	25	293	131.51
n_short_words	2	108	38.83
n_long_words	22	191	92.68
n_sentences	2	23	10.33
n_abbreviations	0	51	9.93
digits	0	215	23.93

Classification

We used four ML models for the binary classification of abstracts into relevant and irrelevant classes at every stage of our approach: random forest (RF), logistic regression (LR), support vector machine classifier (SVC), and naïve Bayes (NB). Each experiment with one ML model applied to a specific set of text statistical features was repeated 300 times with automated resampling to estimate the average classification quality. The number and combination of features were different in distinct tests. In each experiment, the training subset was randomly sampled as 80% of the initial dataset, while the remaining 20% were used as a test subset. Automated sampling stratification was applied using the sklearn python package to keep the subsets class-balanced. The total number of tests performed was 3600 (3 stages x 4 models x 300 resamples).

Results

A total of 630 articles were manually assigned to the relevant (n = 323) and the irrelevant (n = 307) classes prior to experiments. The results within each series of 300 tests were averaged for each ML model combined with a certain set of features (12 combinations).

The average classification quality metrics for the 12 experimental series are demonstrated in Table 2 in descending order of F1-score. The type of ML model is presented in the "ML" column. The "W2V" column shows whether the W2V embedding dimensions were added to the text statistical features. The "KW" column indicated if keyword statistics were used in ML models.

Validation accuracy (accuracy measured on a validation dataset), F1-score, and the area under the receiver operating characteristic curve (ROC AUC) are referred to in the columns "VAC", "F1" and "AUC" accordingly.

Table 2 – Classification quality metrics

#	ML	KW	W2V	VAC	F1	AUC
1	RF	Yes	Yes	0.769	0.775	0.777
2	RF	Yes	No	0.770	0.766	0.768
3	NB	Yes	Yes	0.763	0.763	0.764
4	NB	Yes	No	0.764	0.762	0.764
5	SVC	Yes	Yes	0.741	0.742	0.744
6	LR	Yes	Yes	0.742	0.741	0.744
7	SVC	Yes	No	0.721	0.721	0.725
8	LR	Yes	No	0.720	0.715	0.718
9	RF	No	No	0.669	0.671	0.671
10	SVC	No	No	0.662	0.664	0.665
11	LR	No	No	0.660	0.658	0.658
12	NB	No	No	0.656	0.647	0.649

The best result (F1 = 0.775) was achieved by using an RF model with keywords and W2V components added. RF model without W2V elements achieved F1-score = 0.766. Thus, adding W2V increased the classification quality with the RF model by 1.17%. The inclusion of W2V vectors into the feature space increased F1-score by 0.13% for NB, 2.91% for SVC, and 3.64% for LR.

RF model without keywords and W2V embeddings achieved F1-score = 0.671. Implementation of keyword statistics improved the classification quality by 14.16% compared to the RF model without keywords. For the other models adding keywords increased F1-score by 8.66% for LR, 8.58% for SVC, by 17.77% for NB.

Discussion

Our study focused on a binary classification of short scientific medical texts into user-defined classes, which is more challenging than full-text classification. The classification of articles on full texts might provide significantly better results. However, a real-world practice implies selecting abstracts to be a much more common first-stage process narrowing the literature search to the citations that should be "full-text" screened. Abstracts are obviously much more accessible than full texts.

In this study, we considered the development of classification models through three consecutive stages. The important aspect of our classification approach was its independence of the text content, length, and specificity. We observed that an increasing set of different statistical text properties improves the quality of ML. The best result in the current study was F1-score = 0.775, close to that we obtained previously on word embeddings with the same dataset (F1 = 0.78) [4]. Thus, the classification of short summarized scientific texts grounding on primarily text statistical features can be comparably used as a standalone solution.

We also found that the application of statistical features jointly with word embeddings to a certain extent augments the solution. That is why the presented methods should be considered for further experiments in combination with word vectors and neural language models. Text statistics may additionally contribute to the target variable in the ensemble models.

The impact of keywords for short text classification might be increased using the method described by Y. Gu and J. Shen (2019) [10]. The authors expanded the number of keywords in each short text by similar keywords selected with distance metrics in W2V vector space.

The results of our study demonstrated that the automated classification of scientific abstracts into user-defined classes could be accomplished by using relatively simple and computationally inexpensive methods. This approach allows the implementation of text classification even if only a small dataset is available. It does not require high-performance servers with graphics processing units (GPUs) compared to the BERT technology.

We did not fine-tune hyper-parameters for ML models during the tests, using the defaults provided by the sklearn python package. This allowed us to focus mostly on data preprocessing and feature engineering. We expect that ML models' fine-tuning will improve the classification quality. In our future studies, the new text statistics would be tested and ML models hyper-parameters tuned.

We consider testing new ML models, also in an ensemble design. E.g., such an option might be a high-performance topic memory network (TPM) presented J. Zeng et al. (2020) [11], which enabled multi-label classification of short texts and achieved the best F1-score = 0.964 for Snippets dataset (eight labels) [12], F1-score = 0.851 for TagMyNews dataset (seven labels) [13].

Conclusions

The classification of scientific abstracts might be implemented using relatively simple approaches presented in this paper. These methods are expected to facilitate the selection of literature by researchers, potentially increasing their productivity and research performance.

Acknowledgements

The research was supported by the Russian Foundation for Basic Research grant 19-29-01174.

References

[1] Q.D. Buchlak, N. Esmaili, J.C. Leveque, F. Farrokhi, C. Bennett, M. Piccardi, and R.K. Sethi, Machine learning applications to clinical decision support in neurosurgery: an artificial intelligence augmented systematic review, *Neurosurg. Rev.* **43** (2020) 1235–1253. doi:10.1007/s10143-019-01163-8.

[2] A. Rios, and R. Kavuluru, Convolutional neural networks for biomedical text classification: Application in indexing biomedical articles, in: BCB 2015 - 6th ACM Conf. Bioinformatics, Comput. Biol. Heal. Informatics, 2015: pp. 258–267. doi:10.1145/2808719.2808746.

[3] A.K. Ambalavanan, and M. V. Devarakonda, Using the contextual language model BERT for multi-criteria classification of scientific articles, *J. Biomed. Inform.* **112** (2020). doi:10.1016/j.jbi.2020.103578.

[4] G. Danilov, T. Ishankulov, Y. Orlov, M. Shifrin, K. Kotik, and A. Potapov, The classification of scientific literature for its topical tracking on a small human-prepared dataset, in: Stud. Health Technol. Inform., 2020: pp. 191–194. doi:10.3233/SHTI200526.

[5] G. Danilov, T. Ishankulov, K. Kotik, Y. Orlov, M. Shifrin, and A. Potapov, The classification of short scientific texts using pretrained BERT model, in: Stud. Health Technol. Inform., (accepted, in press), 2021.

[6] F. Abu Sheikha, and D. Inkpen, Automatic classification of documents by formality, in: Proc. 6th Int. Conf. Nat. Lang. Process. Knowl. Eng. NLP-KE 2010, 2010. doi:10.1109/NLPKE.2010.5587767.

[7] F. Zhou, F. Zhang, B. Yang, and X. Yu, Research on short text classification algorithm based on statistics and rules, in: 3rd Int. Symp. Electron. Commer. Secur. ISECS 2010, 2010: pp. 3–7. doi:10.1109/ISECS.2010.9.

[8] G. V. Danilov, M.A. Shifrin, K. V. Kotik, T.A. Ishankulov, Y.N. Orlov, A.S. Kulikov, and A.A. Potapov, Artificial intelligence in neurosurgery: A systematic review using topic modeling. part i: Major research areas, *Sovrem. Tehnol. v Med.* **12** (2020) 106–113. doi:10.17691/stm2020.12.5.12.

[9] G. V. Danilov, M.A. Shifrin, K. V. Kotik, T.A. Ishankulov, Y.N. Orlov, A.S. Kulikov, and A.A. Potapov, Artificial intelligence technologies in neurosurgery: A systematic literature review using topic modeling. Part II: Research objectives and perspectives, *Sovrem. Tehnol. v Med.* **12** (2020) 111–118. doi:10.17691/stm2020.12.6.12.

[10] Y. Gu, and J. Shen, Short Text Classification Based on Keywords Extension, in: Proc. - 2019 Chinese Autom. Congr. CAC 2019, 2019: pp. 2616–2621. doi:10.1109/CAC48633.2019.8996664.

[11] J. Zeng, J. Li, Y. Song, C. Gao, M.R. Lyu, and I. King, Topic memory networks for short text classification, in: Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018, 2020: pp. 3120–3131. doi:10.18653/v1/d18-1351.

[12] X.H. Phan, L.M. Nguyen, and S. Horiguchi, Learning to classify short and sparse text & web with hidden topics from large-scale data collections, in: Proceeding 17th Int. Conf. World Wide Web 2008, WWW'08, 2008: pp. 91–99. doi:10.1145/1367497.1367510.

[13] D. Vitale, P. Ferragina, and U. Scaiella, Classification of short texts by deploying topical annotations, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2012: pp. 376–387. doi:10.1007/978-3-642-28997-2_32.

Address for correspondence

Timur Ishankulov: tishankulov@nsi.ru

Gleb Danilov: glebda@yandex.ru

Laboratory of Biomedical Informatics and Artificial Intelligence, National Medical Research Center for Neurosurgery named after N.N. Burdenko, Moscow, Russian Federation: ai@nsi.ru