AI for AI: What NLP techniques help researchers find the right articles on NLP

Sergei Prokhorov
Moscow Institute of Physics and Technology
Institute for History of Science and Technology RAS
sergei.prokhorov@gmail.com
ORCID ID: 0000-0002-1874-8602

Victor Safronov
Moscow Institute of Physics and Technology
Moscow, Russia
safronov@phystech.edu

Abstract — **The human progress of the coming years is largely associated with success in the field of artificial intelligence methods. The growth of knowledge in this area is in the nature of an information explosion. Researchers have to spend too much time monitoring a constantly evolving field, filtering out the flow of complex documents and texts that require a deep understanding of machine learning methods and their application in specific areas. In solving this problem, the very methods of machine learning and natural language processing are highly effective. The ability to take into account complex non-linear dependencies allows modern language models to effectively solve the problems of information retrieval, monitoring, facts extraction and further analysis. The article provides an overview of modern natural language processing methods that form the basis of modern text information retrieval systems and semi-automatic compilation of reviews and roadmaps. A review of approaches of text vectorization methods for semantic classification of documents. Known limitations of these techniques are also discussed.**

Keywords — information overload; natural language processing; information extraction;

I. Introduction

Since the beginning of the 21st century the rate of accumulation of data in such fields as medicine, geophysics and the data science itself is much faster than the rate of accumulation of interpretation, i.e. the speed with which the scientific community can assimilate data into an interpretive structure [1].

Information overload has caused an increase in the number of studies in the field of information retrieval and natural language processing, as well as the number of accessible information retrieval services. However, due to the commercial specifics of publicly accessible search engines such as Google, Bing, Yahoo, etc., the concept of an "all for all" service has been established, which still does not fully take into account the bibliometric aspects of working with scientific information and provides a meager set of tools to interact with the information found [2]. Another problem of information retrieval remains the closeness of user search "routes", on which the result of the operation of search algorithms depends. This dependence means that the algorithms work in a very complex, proprietary, (unrecordable or even irreproducible and unknowable) context [3].

In this article, we provide an overview of machine learning and natural language processing methods that are fundamental to modern information systems for searching and filtering text documents. They include building a sequence of document processing, generating a feature space, methods for text vectorization, document classification models, and approaches to evaluating the result.

II. Pipeline

At the beginning we have a set of text documents (text corpuses, text corpora). Common way to get a corpora is to download public archives (such as arXiv, PubMed and Wikipedia) and crawling on Internet. Work with a text collection is of a recursive nature, from character-by-character parsing of semantic units of a language (highlighting chunks and morphological analysis of words) to parsing phrases, and so on, to characterizing documents and the entire collection. At the stage of word analysis, the text analysis system should generate a basic set of elements of the attribute space of the model and generate signs for the stage of analysis of phrases and all subsequent stages. At the stage of analysis of phrases, signs are formed both for processing sentences and subsequent stages of processing the entire document. The characteristics formed at the stage of analysis of the proposals are passed on for the analysis of paragraphs and the document.

The next stage is features extraction. Modern approaches to parsing text can solve the problems of classifying and extracting meaning using symbol-by-word text processing (recursive neural networks, attention mechanisms like transformers), and generating feature space from a set of characters (chunking) , words, collocations and n-grams. This step is required to reduce

978-1-7281-4326-2/19/$31.00 ©2019 IEEE
DOI 10.1109/IC-AIAI48757.2019.00023

the dimension and then solve the problems of classification and fact extraction (models with more than a million words require large computing capacities). However, the current models based on attention mechanisms (such as transformers) no longer require efforts to identify a feature space and generate it independently.

We use vector space to represent documents and queries in the information retrieval system. Each document and request is represented by vectors [28]. Each vector representing a document or request consists of a set of features that indicate the words and features generated in the previous steps, and the value of each feature represents the frequency or number of occurrences of that particular word in the document itself. To evaluate the proximity of two documents, the most used is cosine measure of proximity based on vectors assigned to documents. Selection of the best classification algorithm is a highly important step in building a scientific knowledge search system algorithm.

The final step is the selection of a quality assessment method. In practice, the model for assessing the proximity of documents obtained as a result of the above steps is incomplete without taking into account the mechanics of man-machine interaction, however, this nuance does not apply to NLP and is not the subject of this article.

III. Preprocessing and the Feature Space Generation.

Traditional approaches using probabilistic methods involve transforming unstructured text into the space of structured elements and further using classifiers on top of them. But the successes in the NLP of the last rally are associated with combined non-linear models that use character-by-word processing. Methods for parsing sentences at the word level eliminate noise in the data and at the same time generate a feature space for classifying and facts extraction.

Preprocessing

Probabilistic classification algorithms are sensitive to noise in the data. In this case stop words, spelling errors, slang, etc. may turn out to be noise. To avoid this problem the following steps are preliminarily used to prepare the text:

- Tokenization is a preprocessing method that breaks text into words, phrases, characters, or other significant elements called tokens. [8-10].
- Stop words removal. The stage of the necessary reduction of the attribute space. Pronouns, interjections, modal verbs, etc. are removed. Such processing often occurs by setting the word frequency threshold, however, choosing the threshold requires a proper level of skill. [11]
- Casting words to lower case. This is also used to reduce dimensionality due to the fact that the same word can occur both at the beginning and in the middle of a sentence. This method projects all the words in the text and document into the same attribute space, but it causes significant problems for the interpretation of certain words (like “WHO” - World Health Organization and “who” as pronoun). [12]
- Slang, shortenings and abbreviations are the words that require special processing. An abbreviation is an abbreviated form of a word or phrase. E.g. BERT could mean «Bidirectional Encoder Representations from Transformers» as well as hypocoristic form of a number of various Germanic male given names, such as Albert, Herbert, Hilbert etc at the same time.
- Typos correction is important for processing incorrectly compiled search queries and is also present in the data of social networks [14].
- Normalization, stemming and lemmatization. One and the same word may have different meanings (in singular and plural numbers), while the semantic meaning of each form is unique [15]. Stemming removes such elements of words as endings, suffixes (affixes) [16] and is more often used for inflectional languages, e.g. English. Lemmatization is often used to process agglutinative languages (for example, Russian) and is used to bring the word into normal form (lemma) [17].

Common methods for extracting attributes are the term document inverse frequency (TF-IDF), term frequency (TF) [4], Word2Vec [5-6], and global vectors for words representation (GloVe) [7].

Syntactic analysis

The transition from the level of individual words to accounting for joint occurrence allows you to add more syntactic information to the model [19].

- N-Grams. The n-gram technique is a set of n-word which occurs “in that order” in a text set. This is not a representation of a text, but it could be used as a feature to represent a text.
- Syntactic N-Grams. In [20] syntactic n-grams are discussed, which are determined by the tree-like paths of dependencies in the text structure.
- In practice, the attribute space generation by extracting frequent phrases and collocations significantly improves the quality of classification algorithms. One of the most efficient algorithms are TopMine [21] and more advanced AutoPhrase

[22], the difference lies in the complexity of implementation, however both libraries are in the public domain..

Use of Words Frequency

Counting the words occurrence frequency is the dominant concept in building models to compute the proximity of documents using one-hot text encoding.

The Bag-of-Words (BoW) concept is a simplified representation of the text in a document based on the frequency distribution of words, N-Grams and phrases. The technique is used not only in NLP, but also in computer vision. One of the most powerful areas in NLP using this approach is topic modeling [23]. The strong point of using a BoW is the ability to parallelize the learning process of BoW classifiers, the use of many modalities of information (not just text), the ability to build hierarchical classifiers, and the rather high interpretability of the resulting models [24-27]. An obvious drawback of BoW is the lack of word order information, but it only takes seriously the tasks of classifying short texts, which does not apply to scientific publications and popular science texts.

IDF (inverse document frequency) is the inversed frequency with which a certain word occurs in collection documents. This concept is proposed by Karen Spark Jones [29]. IDF accounting reduces the weight of commonly used words. For each unique word within a particular collection of documents, there is only one IDF value. The mathematical representation of the term weight in a document using TF-IDF is given in equation (1):

$$TFIDF(d,w,D) = \frac{n_w}{\sum_k n_k} \times log \qquad (1)$$

Where n_w is the number of occurrences of the word w in the document, $\sum_k n_k$ is the total number of words in the document, $|D|$ - number of documents in the collection, the denominator under the logarithm equals to the number of documents from the collection which contain word w.

Word Embeddings

The syntactic representations of words, including the word bag model, do not take into account the semantics of the word. Their inherent one-hot coding means that their vectors are orthogonal and different words with close meanings will be far in the vector space. This problem is a serious problem for NLP in general. Another problem in the bag-of-word is that the word order in the phrase is also not taken into account.

Words embeddings is a learning technique where each word or phrase from the dictionary is mapped into a vector of dimension N real numbers. The methods of of words vectorization in many respects form the basis of computer linguistics and many works have been devoted to this. The most popular models now are Word2Vec[5-6], GloVe[7], FastText[30], and bulky transformer-based GPT and BERT. All of them use neural networks or deep learning, taking into account the context in a given window. Word2Vec in the basic version is not interpretable, that is, the components of the word vector do not always clearly express one or another aspect of information about entities denoted by words in the dictionary. And vice versa in topic modeling in order to build interpretable models for classifying text documents, matrix factorization and regularization techniques were proposed to give interpreted thematic meaning to the components of vectors [24].

Word2vec (introduced by T.Mikolov[5-6]) accepts a large text corpus as input and maps each word to a vector, giving the coordinates of the words in the output. First, it creates a dictionary, learning on the input text data, and then calculates the vector representation of words. The vector representation is based on contextual proximity: words found in the text next to identical words (and therefore having similar meanings) in the vector representation will have close coordinates of word vectors.

Contextualized word representations are another word embedding technique which is based on the context2vec [номер] technique introduced by B. McCann et al. The context2vec method uses bidirectional long short-term memory (LSTM). M.E. Peters et al. [номер] built upon this technique to create the deep contextualized word representations technique. This technique contains both the main feature of word representation: (I) complex characteristics of word use (e.g., syntax and semantics) and (II) how these uses vary across linguistic contexts (e.g., to model polysemy) [номер]. The main idea behind these word embedding techniques is that the resulting word vectors are learned from a bidirectional language model (biLM), which consist of both forward and backward LMs

FEATURE EXTRACTION COMPARISON

Model	***Limitations***	***Advantages***
Linear Regression on Words	• Requires manual dictionary cleaning • Does not process words order • Polysemy problem is not resolved	• Interpretable • High computational cost • Easy to compute words proximity • Easy to use for novice
TF-IDF	• Does not process words order • No use of words proximity	• Well known interpretable method • Easy to compute documents proximity • Can highlight specific terms in documents
word2vec	• Needs huge text dataset to train (~100k documents)	• Uses position of the words in the text • Uses vector proximity of words

Model	*Limitations*	*Advantages*
Glove	• Not able to resolve polysemy • Storage-consuming • Needs huge corpora to train • Doesn't work with out of vocabulary words	• Works good with huge corpora
ARTM	• Weak for short-text classification • Limited ability of words order accounting	• Very fast when learning • High interpretability, unsupervised learning at the same time • Able to generate hierarchical catalogues • Works with small (~1k) document collections
FastText	• Not able to resolve polysemy • Storage-consuming • Computation is more expensive vs Word2Vec andTopic Models • Low interpretability	• Works on chunks level, applicable for out of vocabulary words

Table 1. Feature generation methods comparison

IV. Classifiers

Logistic Regression and Naïve Bayes

Logistic regression is one of the oldest classifiers, but at the same time the most famous and easily applied. The task of classifiers of this kind is to determine the probability of finding a document in a particular class. Logistic regression is a linear classifier, with the decision boundary:

$$\theta^T x = 0 \qquad (2)$$

Naive Bayes is a generative model. It models the joint *p(x, y)*. However, if our ultimate goal is classification, the relevant part is *p(y|x)*. In Naive Bayes this is computed via Bayes rule. One might wonder whether it is possible to estimate *p(y|x)* directly. A model that estimates *p(y|x)* directly is known as a discriminative model. Logistic regression is one such model.

They both divide the feature space *X* with a hyperplane. Their essential difference is how they find their hyperplane: Naive Bayes optimizes a generative objective function, while logistic regression optimizes a discriminative objective function [31].

KNN

The k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors.

Given a text document d, the k-NN algorithm finds *k* nearest neighbors d among all documents in the training set and evaluates candidates in the category based on the class of k neighbors. The similarity of d and the document of each neighbor can be an estimate of the category of neighboring documents. Criterion of k-NN is

$$f(d_i) = argmax_j S(d_i, d_j) \qquad (3)$$

where we see at document d_i and use some score S to estimate similarity between d_i and subset d_j.

k-NN is easy to implement and works with free choise of feature space. This model also naturally handles multi-class cases. However, k-NN is limited by a capacity of RAM data storage for large datasets. Additionally, the performance of KNN is dependent on finding a meaningful distance function S, thus making this technique a very data dependent algorithm [32].

Support Vector Machines

The idea of the SVM is to find the hypothesis, for which we can guarantee the lowest probability of an error in a randomly selected test case [33]. SVMs are very versatile. In their basic form, SVMs study a linear threshold function. Nevertheless, using a simple change of the corresponding kernel function, they can be used to study polynomial classifiers, radial base function (RBF) networks and three-layer sigmoid neural networks. One of the great features of SVMs is that their learning ability can be independent of feature space dimension. The main limitation of SVM in text classification is time complexity [34].

Decision Trees and Random Forest

One of the popular classification methods (not only for texts) are decision trees [35]. A tree is a branching chain of decision making. Each internal node corresponds to one of the input variables. The tree can also be “studied” by dividing the original sets of variables into subsets based on testing attribute values.

A decision tree can thus be regarded as a source of a message 'P' or 'N', with the expected information needed to generate this message given by

$$I = \frac{p}{p+n} log_2 \frac{p}{p+n} - \frac{n}{p+n} log_2 \frac{n}{p+n} \qquad (4)$$

The main idea is to create a tree, optimally dividing the attribute space in each new node and choosing which attribute or function can be at the parent level and which at the child level. Each child level with its own attribute subspace

$$\mathrm{E} \sum_{i=1}^{v} \frac{p_i+n_i}{p+n} \qquad (5)$$

This way at each node we have to maximize the information gain I - E.

Random Forest [36] is an ensemble learning method which generates random decision trees. Training is based on voting.

Decision trees and random forest quickly learn but simple decision tree works faster because require one pass on the tree when finding a class. But both of these two algorithms are very sensitive to noise in the data, and the problem of imbalanced dataset can lead to overfit. Also, they do not work on data outside the sample set [37].

Probabilistic Topic Modeling

Probabilistic topic modeling is a modern tool for statistical text analysis, designed to identify topics inside collections of documents. The topic model describes each topic with a discrete distribution on a variety of terms, each document describes a discrete distribution on a variety of topics. Topic models are used for information retrieval, classification, categorization, annotation, text segmentation. Topic models allow using not only text but also meta-information as the input, allowing expanding the attribute space.

$$p(w|d) = \sum_{t\epsilon T} p(w|t)p(t \vee d) \qquad (6)$$

The parameters of the topic model - matrices Φ = p(w|t) and Θ = p(t|d) are found by solving the problem of maximum likelihood estimation. However, the two most famous methods for obtaining these matrices, PLSA and LDA[38], give infinitely many solutions of the form (Φ S)*(S^{-1} Θ).

This problem turned to the creation of Additive Regularization for Topic Modeling (ARTM) - a sustainable approach to highlighting the interpretable cluster structure of text collections, not only for IR purposes, but also for subsequent analysis. For this, an effective approach to regularization of matrix decomposition was proposed, which made it possible to construct interpretable hierarchical models. At the output of the model, sparse columns are obtained in which the components of the φ-vectors are their stochastic coefficients that show word's relationship to topics, while the components of the θ-vectors reflect the probability of documents belonging to topics [24]. Model training with ARTM is based on the EM-algorithm and supports parallel computing very well.

V. Conclusion

In this article, we presented a brief overview of the modern methods that are most often used in the practice of creating IR systems that solve the problems of text classification. Also the stage of pre-processing and generation of feature space was highlighted. Indications of the fundamental principles of their work are given and the limitations of existing classification methods are discussed. The main problem in creating information retrieval systems is the balance between available expertise in various text analysis methods and the ability to quickly obtain and interpret model for further evaluation and improvement with given computational power.

References

1. T. Hey, S. Tansley, and K. Tolle, “The Fourth Paradigm. Data-Intensive Scientific Discovery” Microsoft Research. Redmond, Washington, 2009.
2. G. Cabanac et al, “Report on the 8th International Workshop on Bibliometric-enhanced Information Retrieval (BIR 2019)”, arXiv:1909.04954v1, 2019.
3. C. Lynch “Stewardship in the “Age of Algorithms” in First Monday, Vol.22, N.12, 2017
4. G.Salton, C.Buckley “Term-weighting approaches in automatic text retrieval” Inf. Process. Manag. 24, 513–523, 1988
5. Mikolov, Tomas; et al. "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781, 2013
6. Goldberg, Y.; Levy, O. Word2vec explained: Deriving mikolov et al.’s negative-sampling word-embedding method. arXiv, arXiv:1402.3722, 2014.
7. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, Volume 14, pp. 1532–1543, 2014
8. Trim, Craig (Jan 23, 2013). "The Art of Tokenization". *Developer Works*. IBM.
9. Verma, T.; Renu, R.; Gaur, D. Tokenization and filtering process in RapidMiner. Int. J. Appl. Inf. Syst. 2014, 7, 16–18.
10. G.Gupta, Malhotra, S. Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example). Int. J. Comput. Appl. 2015, 975, 8887.
11. Y. Bassil “A Survey on Information Retrieval, Text Categorization, and Web Crawling” Journal of Computer Science & Research

(JCSCR) - ISSN 2227-328X http://www.jcscr.com Vol. 1, No. 6, Pages. 1-11, 2012.

12. Dalal, M.K.; Zaveri, M.A. Automatic text classification: A technical review. Int. J. Comput. Appl., 28, 37–40, 2011.
13. B. Pahwa, S.Taruna, N.Kasliwal, Sentiment Analysis-Strategy for Text Pre-Processing. Int. J. Comput. Appl., 180, 15–18, 2018
14. Mawardi, V.C.; Susanto, N.; Naga, D.S. Spelling Correction for Text Documents in Bahasa Indonesia Using Finite State Automata and Levinshtein Distance Method. EDP Sci. 2018, 164.
15. Spirovski, K.; Stevanoska, E.; Kulakov, A.; Popeska, Z.; Velinov, G. Comparison of different model's performances in task of document classification. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, Novi Sad, Serbia, 25–27 June 2018; p. 10.
16. Singh, J.; Gupta, V. Text stemming: Approaches, applications, and challenges. ACM Compu. Surv. (CSUR) 2016, 49, 45
17. Plisson, J.; Lavrac, N.; Mladeni´c, D. A rule based approach to word lemmatization. In Proceedings of the 7th International MultiConference Information Society IS 2004, Ljubljana, Slovenia, 13–14 October 2004.
18. Caropreso, M.F.; Matwin, S. Beyond the bag of words: A text representation for sentence selection. In Conference of the Canadian Society for Computational Studies of Intelligence; Springer: Berlin/Heidelberg, Germany, 2006; pp. 324–335.
19. Caropreso, M.F.; Matwin, S. Beyond the bag of words: A text representation for sentence selection. In Conference of the Canadian Society for Computational Studies of Intelligence; Springer: Berlin/Heidelberg, Germany, 2006; pp. 324–335.
20. Sidorov, G.; Velasquez, F.; Stamatatos, E.; Gelbukh, A.; Chanona-Hernández, L. Syntactic N-grams as machine learning features for natural language processing. In Expert Systems and Applications 2014; Vol. 41, Issue 3, pp. 853–860.
21. A. El-Kishky Scalable Topical Phrase Mining from Text Corpora http://hanj.cs.illinois.edu/pdf/vldb15_ael-kishky.pdf
22. Jingbo Shang et al. Automated Phrase Mining from Massive Text Corpora https://ieeexplore.ieee.org/document/8306825
23. D.M. Blei. ”Probabilistic topic models”, Communications of the ACM, 55(4), 2012, pp. 77-84.
24. K.V. Vorontsov and A.A. Potapenko. ”Additive regularization of topic models”, Machine Learning, Special Issue on Data Analysis and Intelligent Optimization with Applications, 2015, 101(1), pp. 303-323.
25. N.A. Chirkova and K.V. Vorontsov. ”Additive regularization for hierarchical multimodal topic modeling”, Journal Machine Learning and Data Analysis, vol.2(2), 2016, pp. 187-200
26. O.Frei and M.Apishev. ”Parallel non-blocking deterministic algorithm for online topic modeling”, in AIST2016, Analysis of Images, Social networks and Texts, vol.661, 2016, pp. 132-144. Springer International Publishing Switzerland, Communications in Computer and Information Science (CCIS).
27. Anastasia Ianina, Konstantin Vorontsov Regularized Multimodal Hierarchical Topic Model for Document-by-Document Exploratory Search, unpublished
28. Gerard Salton, A. Wong, C. S. Yang, “A Vector Space Model for Automatic Indexing”, CACM 18(11), 1975.
29. Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. J. Doc. 1972, 28, 11–21.
30. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. arXiv 2016, arXiv:1607.04606.
31. Qu, Z.; Song, X.; Zheng, S.; Wang, X.; Song, X.; Li, Z. Improved Bayes Method Based on TF-IDF Feature and Grade Factor Feature for Chinese Information Classification. In Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 15–17 January 2018; pp. 677–680
32. Jiang, S.; Pang, G.; Wu, M.; Kuang, L. An improved K-nearest-neighbor algorithm for text categorization. Expert Syst. Appl. 2012, 39, 1503–1509.
33. Vladimir N. Vapnik. The Nature of Statistical Learning Theory. Springer, New York, 1995
34. Zhang, W.; Yoshida, T.; Tang, X. Text classification based on multi-word with support vector machine. Knowl.-Based Syst. 2008, 21, 879–886.
35. Quinlan, J.R. Induction of decision trees. Mach. Learn. 1986, 1, 81–106.
36. Breiman, L. Random Forests; UC Berkeley TR567; University of California: Berkeley, CA, USA, 1999
37. Giovanelli, C.; Liu, X.; Sierla, S.; Vyatkin, V.; Ichise, R. Towards an aggregator that exploits big data to bid on frequency containment reserve market. In Proceedings of the 43rd Annual Conference of the IEEE Industrial Electronics Society (IECON 2017), Beijing, China, 29 October–1 November 2017; pp. 7514–7519.
38. Blei D. M. Probabilistic topic models // Communications of the ACM. 2012. Vol. 55, no. 4, pp. 77-84.