Automatic Classifier of Scientific Contents

Samuel Machado[0000-0002-9078-5280] and Jorge Oliveira e Sá[1][0000-0003-4095-3431]

[1] Algoritmi Center, University of Minho, Braga, Portugal

Abstract. The growth of scientific production, associated with the increase in the complexity of scientific contents, makes the classification of these contents highly subjective and subject to misinterpretation. The taxonomy on which this classification process is based does not follow the scientific areas' changes. These classification processes are manually carried out and are therefore subject to misclassification. A classification process that allows automation and implements intelligent algorithms based on Machine Learning algorithms presents a possible solution to subjectivity in classification. Although it does not solve the inadequacy of taxonomy, this work shows this possibility by developing a solution to this problem. In conclusion, this work proposes a solution to classify scientific content based on the title, abstract, and keywords through Natural Language Processing techniques and Machine Learning algorithms to organize scientific content in scientific domains.
Keywords: Taxonomy, Machine Learning, Natural Language Processing

1 Introduction

Humans learn to classify things at a very young age. Categorizing fills a need of human nature, that is, to impose order and find hidden relationships. However, we are not very good at classifying because we organize empirically, based on intuition or experience. It is simple to classify a set of ten black and white balls into two classes: black and white. But as we increase the number of characteristics, so does the complexity of the task. Classification allows us to understand diversity better.

A Text Classifier is an abstract model, which describes a set of predefined classes generated from a collection of labeled data or training set. The classifier is used to correctly classify new texts for which the class label is unknown [1].

Real-world raw data is usually unsuitable for direct use in classifier training, so some cleaning and preprocessing steps are generally applied before the classification task. Thus, scientific contents must go through a Natural Language Processing (NLP) techniques for the data to be ready for classification [2].

Classification in science adds several challenges, some of which can result in biased models when we try to understand feature like:

- The actual content of the document. It is sometimes classified into an existing class even when it does not fit in an emerging research field.

- The person that decides the classification can be either the author, the designated person who submits the publication, or a committee of peers.

With the increase in publications, the human factor, especially under the pressure of numbers or information overload, is most likely to make mistakes and fail to identify correctly and consistently. Humans often prone to errors during analysis or when trying to establish relationships between multiple features. Machine Learning algorithms can be applied to solve or mitigate these problems while improving efficiency.

2 Concepts and Subjects

To provide a suitable solution to the problem under study, we needed to address some concepts and subjects. Regardless of the classification system, the variety of Machine Learning (ML) classification techniques is wide and constitutes this core. Thus, in this section, we will address two: classification systems and how to develop automatic classifiers based on ML algorithms.

2.1 Classification Systems

As scholarly research becomes increasingly interdisciplinary, an essential purpose for a classification system is to facilitate multidisciplinary research and information sharing [3]. Comte [4] proposed a schema of science classification. He argues that the division of intellectual labor is necessary and that the scientific domains would have to be cultivated separately. He also stressed that the sciences all belonged to a larger whole and that any division is artificial.

A classification system should contain, amongst other features [5]:

- Breadth - defined as either a typology or a taxonomy based on classes where the subjects would be classified or grouped;
- Meaning - supporting the rational use of the selected classification method and classes should be a philosophical foundation;
- Depth – as close as possible to support the diversity of real-life phenomena;
- Recognizability – must mirror the real world.

To better understand a classification system, we need to understand the concepts of taxonomy, ontology, and thesaurus [6] finally, how it can be applied to the classification of science results (for example, articles). For example, taxonomy allows to define groups of biological organisms based on shared characteristics and to name these groups. Thus, it groups the organisms in a taxonomic classification; groups of a given class can be aggregated to form a higher-level supergroup, thus creating a taxonomic hierarchy [7]. A taxonomy typically has some hierarchical relations incorporated in its class classifications. Thesaurus can be understood as a taxonomy extension: it takes taxonomy as described above, allowing subjects to be arranged in a hierarchy. Besides, it adds the ability to enable other statements to be made about the topics. Both the taxonomy and the thesaurus can fall into the Knowledge Organization Schemes (KOS)

class because they provide the set of structured elements to be used for describing and indexing objects, browsing collections, etc. Ontology, originally from the philosophical domain, has been given a new definition with the development of Artificial Intelligence as a formal, explicit specification of a shared conceptualization [8]. They represent the set of objects, properties, and relationships we can use in a specified domain of knowledge. By defining the terms and their relationships, ontology encodes a knowledge domain so that a machine can understand it. The W3C standard for defining ontologies is OWL, a key component of semantic web technologies [9]. Ontologies are also often interpreted as the classification mechanism itself. A controlled vocabulary is a closed collection of terms that have been explicitly grouped and can be used for classification. It is controlled because the list is limited, and there is control over who can add terms to the list, when, and how (Fig 1).

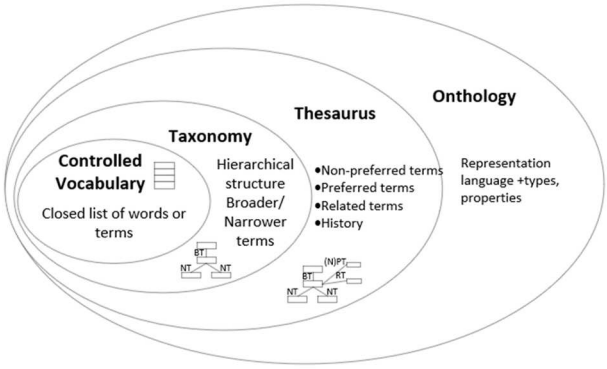

Fig. 1. - Classification Categories, adapted from [10]

2.2 Machine Learning

Artificial Intelligence (AI) can be used in Texts and Knowledge Discovery Databases using NLP techniques. For example, this serves to annotate automatically, and index texts through text corpora classification, which requires external data support in the form of ontologies, thesaurus, etc. [11]. However, there are restrictions on applying new patterns not yet discovered, often in innovative scientific publications [12].

ML aims to provide automated extraction of insights from data. Standard learning systems (like neural networks or decision trees) operate on input data after they have been transformed into feature vectors. The data vectors or points can be separated by a surface, clustered, interpolated, or otherwise analyzed. The resulting hypothesis will then be applied to test points in the same vector space to make predictions or classifications [13]. This approach loses all the word order information, only retaining the terms' frequency in the document by removing non-informative words (stop words) and replacing words with their stems or stemming [14]. NLP, in its many aspects, is illustrated in Figure 2. On the left side are represented the requirements to develop an NLP system. The first big challenge is to get enough data as a word dictionary to provide the

system with enough linguistic and semantic knowledge of each possible class in taxonomy to use.

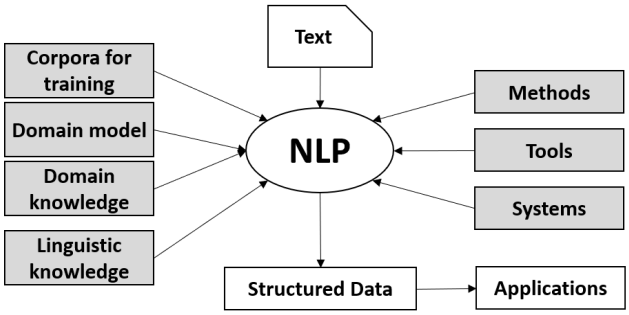

Fig. 2. - Aspects of NLP, adapted from [15].

The right side of Figure 2 represents NLP's operationalization with the methods, systems, and tools. The output with structured data can then feed an ML (or other) system. We can split natural language understanding at a word level, and concept level approaches as Syntax-centered NLP and Semantics-based NLP, respectively. NLP has excellent potential to be used as a preprocessing step on a classification ML or classifier itself. Recent investigations show that NLP's use as a pre-processor for neural networks or in a more advanced fashion Convolutional Neural Networks, with multiple levels and stages of perceptrons[16], and supported by a Thesaurus and a useful Ontology can achieve good classification results. There would still be some limitations for the discovery of new classes, though. This preparation of the texts is relevant to apply a Taxonomy capable of dealing with science's complexity, i.e., scientific documents present interdisciplinarity of scientific domains. [17]. Thus, the classification of scientific documents includes an additional complexity factor in applying a scientific taxonomy [15].

3 Application Scenario

The ALGORITMI Research Center is a research unit of the University of Minho, Portugal, that develops R&D activity in Information and Communications Technology and Electronics (ICT&E) and it is divided into four research fields [18]:

1. Electrical Engineering, Electronics, and Nanotechnology.
2. Operations Research, Statistics, and Numerical Methods.
3. Information Systems, Software, and Multimedia.
4. Communications, Computer Networks, and Pervasive Computing.

ALGORITMI includes 9 R&D groups, divided into 14 R&D domains, the number of integrated researchers at Algorithm Center is 102, but the total number of researchers (integrated and collaborators) are approx. 500.

We can start to ask if the taxonomy in place can deal with these multidisciplinary publications. ALGORITMI internally uses a taxonomy equivalent to that adopted by

governmental institutions of science, as one would expect. Which in turn follows a taxonomy recommended by the OECD, called the Frascati Manual. This taxonomy suffers from reduced depth levels, tending to generalize more and, therefore, to be somewhat limiting or reductive, causing an increase in overlap or high aggregation of domains or subjects.

The scientific publications, produced in ALGORITMI, cover the four research domains and the 14 existing R&D domains. The increase in the number of coauthors per publication may or may not belong to different R&D domains, leading to a rise in publications belonging to various scientific research communities, causing an increase in publications covering several R&D domains. For example, the scientific article produced in ALGORITMI "Calado, A., Leite, P., Soares, F., Novais, P., & Arezes, P. (2018). Design of a Framework to Promote Physical Activity for the Elderly. In International Conference on Human Systems Engineering and Design: Future Trends and Applications (pp. 589-594). Springer, Cham." apparently belongs to the scientific domain linked to Health, but the article reports the development of a UI that allows to show and compute real-time results of the Boccia game. From here, two points are clear:

- Cross-domain research, which shares disciplinary knowledge by investigating a phenomenon, presents additional complexity to the classification system;
- A classification method requires an increasing effort to maintain consistency to cope with existing complexity, making existing classification systems unable to allow correct classifications.

The complexity and dynamics of science make existing taxonomies, which, as a rule, are static, into inaccurate classification results. To make taxonomies more dynamic, i.e., the ability to arise new disciplines through an iterative interdisciplinarity cycle [19].

The problems identified in the Manual Frascati taxonomy were also verified in other studied taxonomies, e.g., Scopus, Microsoft Academic, CORDIS, among others. Classification inconsistencies, different approaches, and scalability are some of the additional problems identified. Therefore, from the results obtained in the taxonomies analysis process, it was possible to accomplish a Frascati Manual taxonomy adaptation with major identified problems fixed. This adapted taxonomy consists of 15 scientific knowledge domains and 447 scientific knowledge subdomains and was implemented in the developed classification system.

4 An Automatic Classifier

The hardware used for this study, namely to training the classification algorithms, was a CPU Intel(R) Core(TM) i7-7700HQ 2.80GHz, with 16GB of RAM and a 250GB SSD, and the dataset used in this study contains scientific publications produced by ALGORITMI researchers. In total, there are 2,665 scientific documents created between the years 2008 and 2017. Of these 2,665 documents, 2,389 are coauthored. All these documents were classified manually by a librarian by using the Frascati Manual

taxonomy. Therefore, the documents were manually reclassified according to the adapted taxonomy.
Table 1 presents the structure of the dataset. It contains ten fields. The goal is to classify the fields "knowledge domain" and "knowledge subdomain", and the training set includes the previous manual reclassification.

Table 1. - ALGORITMI dataset fields

#	Field	Sample
1	Author	S. Azevedo
2	Publication	Systematic Use of Software Development Patterns through a Multilevel and Multistage Classification
3	Type of publication	Book Chapter
4	Knowledge Domain	Computer and Information Science
5	Knowledge Subdomain	Computer Sciences
6	Date of publication	2011
7	Weblink	https://www.scopus.com/record/display.uri?eid=...
8	Coauthors	A. Bragança; R. J. Machado, H. Ribeiro
9	Abstract language	English

Python programming language is becoming very popular in ML applications. The justification is because Python includes several ML libraries, and there are packages ready to use, for instance, Anaconda. It turns out that we can find some top-rated scientific computing tools, including Deep Learning (DL) virtual environments. Anaconda provides integrated end-to-end tools to manage libraries, dependencies, and environments to develop and train ML and DL models and analyze data, including data visualization tools. Through Anaconda, the Jupyter Notebook served as a virtual Python environment, and Python 3 kernel (version 3.7.4) was used for this task. Because it provides easy-to-use APIs for a wide variety of text preprocessing methods, Python's Natural Language ToolKit (NLTK) was installed, providing predefined NLP tasks. It is one of the most used libraries for NLP and computational linguistics. It consists of a suite of program modules, data sets, and tutorials supporting research and teaching in computational linguistics and NLP. NLTK contains several corpora and includes a small selection of texts from Project Gutenberg, which provides 25,000 free electronic books. The toolkit Stopwords Corpus package enables the remotion of redundant repeated words. To do data analysis, the platform used is Pandas. Pandas provide high-performance, easy-to-use data structures and data analysis in Python programming language, allowing fast analysis and data cleaning and preparation. Pandas' alternative would be Numpy or Scipy, but Pandas works well with labeled data, hence the root of Pandas name: Panel Data. Numpy could be more helpful for the numerical data type (Num).

We need to disambiguate the meanings of the sentences by eliminating the punctuation. It introduces noise and adds little value to the analysis capacity based on a text's

word vectors, which in this study case. The punctuation is removed by running a function through each character in the sentence and removes it. Removing punctuation from a text makes it unstructured. The tokenization process separates this text into units, such as phrases or words, by giving structure to a previously unstructured text. For example, the sentence "Modeling Software Product" is divided into tokens [modeling, software, product]. This task is useful to prepare the text to be handled by a lexical analyzer, which is the next step. After the text's tokenization, we can feed a lexical analyzer to remove "stop words". These are generally the most common words used in a given language and do not add any value to the data. The NLTK contains a list of irrelevant words in English, so it is necessary to process the text using a lexical analysis function that compares each word with the items in this list and removes them. The remaining text was properly tagged using Part-of-Speech tagging and since it still contains several derived words two approaches can be followed: Stemming, to eliminate words inflected (or sometimes derived) to the word stem, base, or root form. This is useful for simplifying words in the text without losing their meaning (except in a semantic analysis, which is not the case); Lemmatization reduces the words "modeling", "modeled", and "modeler" to the root word, "model". We find that Stemming's approach cuts the end of words. In this way, the words are meaningless as "sourc" or "emiss". Although the process is fast, it is not very useful and can reduce the model's accuracy. On the other hand, the Lemmatization approach is based on a dictionary to make a morphological analysis of the word to determine its root form.

After text processing, the next step is to test a collection of classifiers to assess the speed and accuracy of each algorithm used. For all the algorithms used, the resulting models will be built based on vectorization data. A TF-IDF is applied, and the relative count of each word is stored in a sparse matrix. TF-IDF differs from the standard TF calculation that counts only the frequency of terms and would give more weight to longer documents than shorter documents. The IDF calculates the term frequency times the inverse frequency of the document. For the algorithms training and testing process, the data was split into two different blocks, the training block having 70% of the total data and the testing block having the remaining 30%. To the initial data, it has added the abstract text of the article, extracted from the location "weblink" in the original dataset. One possible approach for using ML to classify documents could be the author field. In a scarce dataset, the model is highly biased by the author's affiliation to a particular school or domain. Therefore, the author's names were disregarded as classification features. It is possible that, with a better dataset, the attributes author and affiliation can be used to improve the accuracy of the model. Therefore, given the low quantity of data available and the fact that several potential good features were ignored with the intent not to influence the model (like author's name or affiliation), the scores were very promising, with NB and SVM models scoring 80% accuracy. However, the obtained accuracy was also achieved since the data used to train the algorithms was unbalanced, which resulted in a biased model. Hence, it was necessary to proceed to the data balancing resorting to oversampling technics to verify if the models accuracy will improve. After the data balancing process, were also implemented features to optimize the hyperparameters of the ML algorithms automatically, using the GridSearchCV module

from the sci-kit learn library. Therefore it was selected a set of values for each hyperparameter for each ML algorithm to allow the optimization module to find the optimal set of hyperparameters.

This work used the algorithms: Support Vector Machine (SVM), namely SVC, LinearSVC, and NuSVC; Naïve Bayes (NB), specifically MultinomialNB, BernoulliNB, and ComplementNB; and Neural Network (NN) using the MLP classifier. To make the comparison between algorithms were performed one hundred hyperparameters optimizations for each algorithm. Thus, metrics related to the precision of the algorithms were collected, namely, the optimization score and elapsed time. Figure 3 presents the accuracy and optimization score of the used algorithms. Table 2 shows the elapsed time divided into four columns: the first two columns contain the training time of the best model achieved and the average training time of the algorithms, and the remaining columns present the average and the total time of the algorithms hyperparameters optimization.

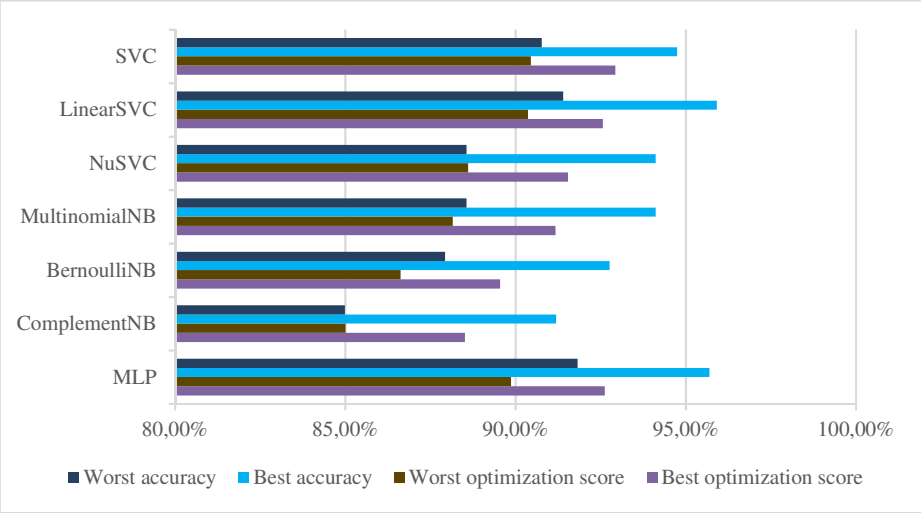

Fig. 3. – Accuracy achieved with the algorithms adopted

Table 2. Algorithms training and optimization time comparison

Algorithms	Best model training time (s)	Average training model time (s)	Average optimization time (s)	Total optimization time (m)
SVC	33.51	33.67	135.73	222.66 (≈ 3.71 h)
LinearSVC	0.07	0.06	28.26	47.42
NuSVC	2.32	2.36	115.38	191.7 (≈ 3.2 h)
MultinomialNB	≈ 0.002	≈ 0.002	0.69	1.22
BernoulliNB	≈ 0.003	≈ 0.003	0.79	1.39
ComplementNB	≈ 0.002	≈ 0.002	0.80	1.40
MLP	16.98	16.80	484.58	844.36 (≈ 14 h)

5 Conclusions

The algorithm with the best accuracy result obtained was the LinearSVC algorithm, belonging to the class of SVM algorithms, with an accuracy of 95.91%. In the class of NB algorithms, the MultinomialNB algorithm reached an accuracy of 94.12%, and the MLP algorithm, belonging to the NN algorithms class, got the second-best precision value in the total set of ML algorithms with 95.70%.

However, the training time is also a relevant factor in the algorithm implementation, since, ideally, the implemented algorithms should be able to learn continuously. Therefore, depending on the requirements of the implementation, it necessary to consider if it is worth it, a higher training time for better accuracy. For example, the training time of the best accuracy MLP classifier took 16.98 seconds while to train the best MultinomialNB classifier took 0.002 seconds, which means that for an accuracy improvement of 1.58%, the time needed to train got 8490 times higher. With this low amount of data, the time difference is already substantial, but with the continuous learning of the algorithms, the training time could get unbearable.

The training time of the algorithms, to be able to make an automatic classification with high accuracy, can be long. Still, the time necessary for manual classification of scientific contents is much more significant and subject to errors. Thus, the need arises to verify the result of automatic classifications with the result of manual classifications. Therefore, it was verified whether the "wrong" classifications made by the algorithms to the test dataset were analyzed to understand if they were wrong or if the scientific content was manually classified in the wrong way.

An ML text classifier based on supervised learning is highly dependent on the amount of training data available. The results obtained in this work can improve with the increase in the amount of training data, as well as in terms of quality. For example, authors identification, authors affiliation were not used for this purpose. Another attribute relevant is the keywords, but it would be useful to use keywords supported on controlled vocabulary from a taxonomy. However, a future automatic classifier tool should validate the keywords through an ML algorithm to detect emerging areas of knowledge or alert for misuse of keywords. To increase the classification accuracy, we propose to editors (conferences and journals) to limit the keywords used in an article to a controlled vocabulary based on taxonomic classes.

To be explored is also the integration of more complex ontology-based knowledge in classification. The development of more efficient non-associative classification algorithms that integrate taxonomy information in classifier training and DL's use, the more data you give and the more computational time you provide, the better accuracy classification is obtained.

Finally, there is a need to classify into multiple knowledge domains correctly, and a classification tool must consider this.

Acknowledgments

This work has been supported by IViSSEM: POCI-01-0145-FEDER-28284.

References

1. Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging Artificial Intelligence Applications in Computer Engineering, 160, 3–24.
2. Romanov, A., Lomotin, K., & Kozlova, E. (2019). Application of natural language processing algorithms to the task of automatic classification of Russian scientific texts. Data Science Journal, 18(1), 1–17. https://doi.org/10.5334/dsj-2019-037
3. Jones, K. S. (2005). Some thoughts on classification for retrieval. Journal of Documentation, 61(5), 571–581.
4. Comte, A. (1988). Introduction to Positive Philosophy. Hackett Publishing.
5. Rich, P. (1992). The Organizational Taxonomy: Definition and Design. The Academy of Management Review, 17(4), 758–781. https://doi.org/10.2307/258807
6. Brewster, C., & Wilks, Y. (2004). Ontologies, Taxonomies, Thesauri: Learning from Texts. 32.
7. Frodeman, R., & Klein, J. T. (Eds.). (2012). The Oxford handbook of interdisciplinarity. Oxford Univ. Press.
8. Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. Data & Knowledge Engineering, 25(1–2), 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6
9. *OWL - Semantic Web Standards*. (2012, December 11). https://www.w3.org/OWL/
10. Kopácsi, S., Hudak, R., & Ganguly, R. (2017). Implementation of a classification server to support metadata organization for long term preservation systems. Mitteilungen der Vereinigung Österreichischer Bibliothekarinnen und Bibliothekare,70(2), 225-243.
11. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. AI Magazine, 17(3).
12. Atkinson-abutridy, J., Mellish, C., & Aitken, S. (2003). A semantically guided and domain-independent evolutionary model for knowledge discovery from texts. IEEE Trans. Evol. Comput, 546–560.
13. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. Journal of Machine Learning Research, 2(Feb), 419–444.
14. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. European Conference on Machine Learning, 137–142.
15. Friedman, C., Rindflesch, T. C., & Corn, M. (2013). Natural language processing: State of the art and prospects for significant progress, a workshop sponsored by the National Library of Medicine. Journal of Biomedical Informatics, 46(5), 765–773. https://doi.org/10.1016/j.jbi.2013.06.004
16. D. W. Otter, J. R. Medina, and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," 2019.
17. M, P., Ozkan, S., Wang, H., & Bridges, S. M. (2012). Cross-Ontology Multi-level Association Rule Mining in the Gene Ontology.
18. ALGORITMI – Universidty of Minho. (2020). http://algoritmi.uminho.pt/
19. Frodeman, R., & Klein, J. T. (Eds.). (2012). The Oxford handbook of interdisciplinarity. Oxford Univ. Press.