An Automatic Text Classification Method Based on Hierarchical Taxonomies, Neural Networks and Document Embedding: The NETHIC Tool

Luigi Lomasto[1], Rosario Di Florio[2], Andrea Ciapetti[3], Giuseppe Miscione[3], Giulia Ruggiero[3], and Daniele Toti[3,4](✉)

[1] Eustema S.p.A., Naples, Italy
l.lomasto@eustema.it
[2] Allianz SE, Munich, Germany
rosario.di-florio@allianz.com
[3] Innovation Engineering S.r.l., Rome, Italy
{a.ciapetti,g.miscione,g.ruggiero,d.toti}@innen.it
[4] Department of Sciences, Roma Tre University, Rome, Italy
toti@dia.uniroma3.it

Abstract. This work describes an automatic text classification method implemented in a software tool called NETHIC, which takes advantage of the inner capabilities of highly-scalable neural networks combined with the expressiveness of hierarchical taxonomies. As such, NETHIC succeeds in bringing about a mechanism for text classification that proves to be significantly effective as well as efficient. The tool had undergone an experimentation process against both a generic and a domain-specific corpus, outputting promising results. On the basis of this experimentation, NETHIC has been now further refined and extended by adding a document embedding mechanism, which has shown improvements in terms of performance on the individual networks and on the whole hierarchical model.

Keywords: Machine learning · Neural networks · Taxonomies · Text classification · Document embedding

1 Introduction

The last decade has seen an extremely high surge in the usage of network-based technologies by people in their everyday lives, and as such an enormous amount of information, a significant part of it in textual form, is being exchanged at a constant rate. As a matter of fact, social networks and online platforms are now an essential way for people to share documents and data, but at the same time they are leading to an increase of confusion and of potentially hidden

L. Lomasto and R. Di Florio—Contributed equally to this work.

or lost information. In order to put some measure of order upon this deluge, methods and techniques have arisen to try and provide users with means to classify the textual information exchanged in a manner that may be as automatic as possible. This is critical for an effective management and exploitation of the information itself. Considerable efforts have been spent so far to solving this problem, by bringing about corresponding solutions for text classification both in literature and in commercial platforms [7].

In this regard, machine learning techniques such as supervised classification can be effectively used to assign a number of predefined labels or classes to a given textual document [19].

This work describes NETHIC, a software tool implementing an automatic text classification method which, at its core, relies upon hierarchical taxonomies and artificial neural networks (ANNs). The tool had been earlier introduced in [6] in its original form, where its cross-domain applicability had been shown by detailing an experimentation on both a general purpose and a domain-specific classification task. The latter revolved around a European funded project for detecting and analyzing criminal contents from online sources.

In this paper, NETHIC's core elements and features are firstly reprised, and then an extension to its methodology and functionality is described, which is meant to improve the overall performance of the neural networks by means of a Document Embedding mechanism, as proposed by Google [15].

Afterwards, a corresponding additional experimentation process is reported, showing the expected improvements in terms of performance both on the individual networks and on the whole hierarchical model.

The structure of this work is the following. In Sect. 2, related work is discussed. Section 3 summarizes the core elements making up NETHIC and provides a brief introduction to the Doc2Vec mechanism introduced to enhance it. Section 4 describes the extensions of NETHIC's original framework by detailing its current architecture, pre-processing, training and core algorithms used. Section 5 reports a new experimentation showing a comparison between NETHIC's original performance and the one resulting from the enhanced method. Section 6 finally concludes the work and hints at future developments.

2 Related Work

Classifying textual sources (documents) is a complex task that can be tackled by computational methods like text mining and natural language processing. These methods, as applied to different corpora and domains, include document conceptualization and summarization [23], subject categorization [19], sentiment analysis and author recognition [14,28], and so forth. The classical approach shared by a number of the aforementioned methods for classifying texts is to represent them via high-dimensional vectors of features, to be passed to machine-learning classifiers [24,27]. Vectors of features are built via a range of different techniques [9], but the most common relies upon using frequencies of specific words or sets of words (like n-grams, phrases, etc.) featured in documents within a corpus. These frequencies can potentially be weighted as well. This technique is commonly known as bag-of-words (BOW): typically, in such a technique keywords are derived from training data, and a plethora of NLP methods is applied in order to do that, including POS Tagging, Named Entity Recognition, Relation Detection and others [1–4,22].

In literature, different methods have been proposed to achieve optimal classification performance. For example, Naive Bayes demonstrates the effectiveness and efficiency for classifying test documents [16,17], but it has poor performance when some categories are sparse [20]. As one of the deep learning algorithms, recurrent neural network (RNN) is proposed by Pyo and Ha to deal with the multi-class classification problem with unbalanced data [10], in which the learnt word embedding depends on a recursive representation of the same initial feature space. In addition, convolutional neural network (CNN) achieves remarkable performance in sentence-level classification [12,13,31]. Recently, CNN has been regarded as a replacement for logistic regression models [32], which uses pre-trained word vectors as inputs for training the CNN models [32]. NETHIC therefore took this line as its preferred choice, in order to achieve the best possible synthesis between correctness of results and general performance of the classification system. Hierarchical classification suffers from error propagation issue [21]. Zhou et al. [33] show that only oversampling and threshold moving is effective for training cost-sensitive neural networks by empirical studies. However, it becomes difficult to define costs of misclassification when there are large number of classes. A more recent paper [5] also supports similar claims, but suggests to use NLP and vector representation of sentences and words as a key, to reduce the propagation of errors between categories in the hierarchy. Choosing this approach, NETHIC and, in particular the extended version of the tool, has chosen as a basis for the calculation of similarities a bag-of-word approach and a multi-dimensional vector analysis of the structure of sentences expressed in natural language. This allowed for a discreet but still significant improvement over the previous version.

3 Core Elements of NETHIC

This section summarizes NETHIC's core elements, by providing a brief description of NETHIC's main approach, and by detailing the structure of the hierarchical taxonomy needed for the tool to work and the datasets used in the training phase.

3.1 NETHIC's Approach

Taxonomies are data modeling structures able to describe knowledge in a way that is both machine-processable and human-friendly. Their inner hierarchy can easily serve the purpose of supporting the process of classifying contents either for a human user or for a computational mechanism [29]. Besides, ANNs are a class of machine-learning methodologies that has proven to be significantly useful for discovering patterns among resources.

A combination of ANNs and taxonomies can therefore be extremely effective when dealing with huge amounts of data, and can also scale pretty well on multi-processor or multi-core hardware architectures, outperforming in terms of processing time several other types of mechanisms sharing comparable levels of effectiveness [11].

This is exactly the approach used by NETHIC and the objective it tried to achieve.

The earlier validation of the soundness and effectiveness of NETHIC's method had been carried out on a corpus made up of Wikipedia articles representing subject categories, including 500 articles for each category [6]. Mechanisms used for the tokenization process include NLTK (Natural Language Processing Toolkit) algorithms, whereas for extracting features, instead of encoding the frequency of keywords, sentences have been decomposed into words, and the latter have been turned into sequences of vectors and then passed to the deep learning methods. In this regard, a certain similarity is shared between NETHIC's approach and other models based on probabilities, including Markov models, conditional random fields and n-grams.

3.2 Underlying Data Model

As described in [6], NETHIC's starting point lies in the use of taxonomies, since their very nature as a hierarchical representation contributes to place a certain level of order on top of unstructured or semi-structured textual data. As such, with their clearly-defined logical categories, they also make accessing and browsing such data dramatically easier both for users and for software systems.

As a matter of fact, taxonomies play an essential role within NETHIC because they are used as a bridge meant to connect its underlying knowledge model (with data taken from Wikipedia) to subsequently build the datasets used during the training phase of the method. These tree-like structures were shown to provide the output and input classes for the respective input and output layer of each neural network used in NETHIC, demonstrating their critical role in the training phase [6].

While the research described in [6] reported the use of two taxonomies, with one of them having the purpose of tackling a domain-specific classification task, this work now focuses only on a general-purpose taxonomy that covers a wide range of general topics, whose upper and lower halves are displayed in Figs. 1 and 2, respectively.

This taxonomy is defined in the RDF standard [26], starting from an abstract classification class, here referred to as *root*, and moving to the lower class, here referred to as *leaf*. This classification structure contains 21 root child categories with a tree depth of 2, with a total of 117 leaf categories. Not all the categories feature an expansion of the tree structure, *i.e.* not every sub-tree of the taxonomy possesses the same structure in terms of the sub-categories of the classification.

Each leaf is then manually connected to the associated category of the Wikipedia graph by using the SKOS properties `skos:exactMatch` and `skos:majorMatch` [25], which support the subsequent construction of the dataset explained in Sect. 3.3.

3.3 Dataset

NETHIC currently makes use of a dataset of 57,304 text documents, taken from Wikipedia. Wikipedia was chosen for this research because it provides an extensive general-purpose knowledge archive that is regularly updated, as well as being easily accessible on the Internet via its HTTP APIs. As mentioned earlier, the

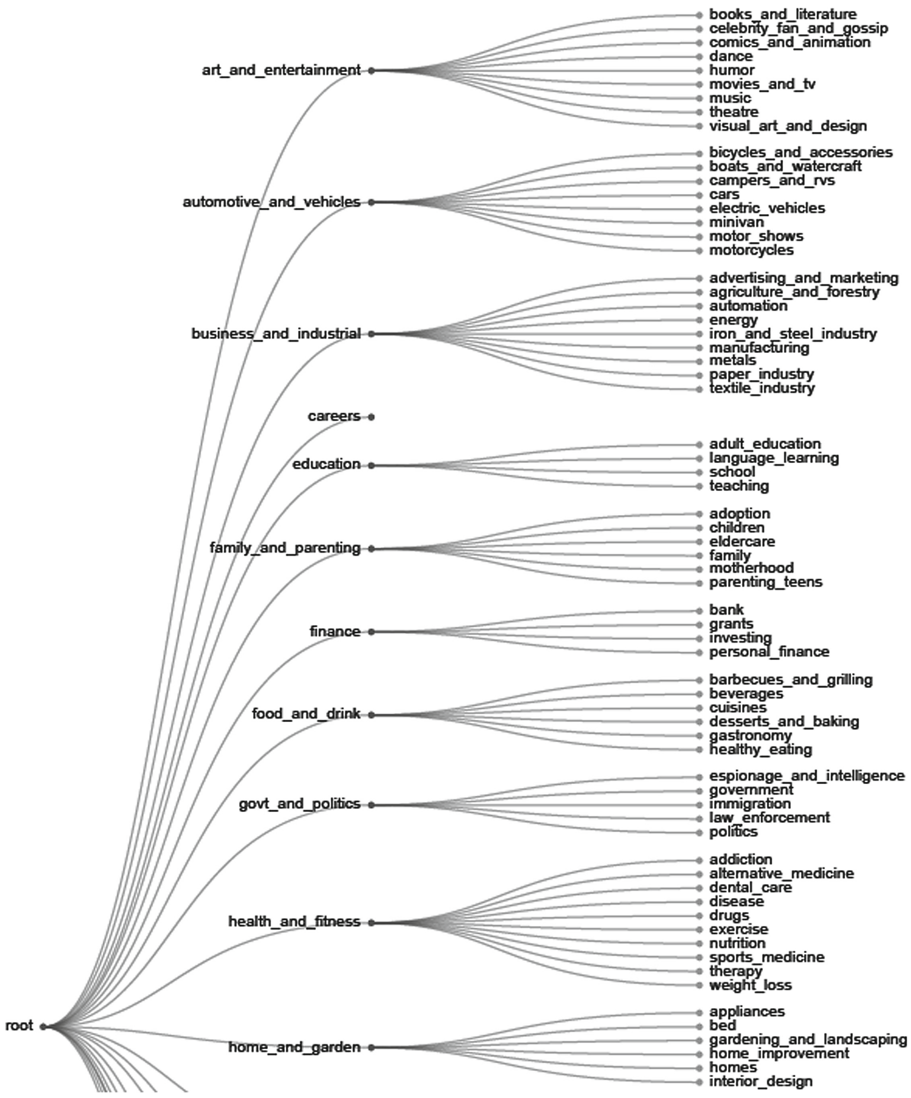


Fig. 1. General-purpose hierarchical taxonomy (upper part).

taxonomy defined as NETHIC's core knowledge model is used to provide the connections between the various classes and the Wikipedia knowledge graph. The first step of this process lies in downloading the entire library of categories from Wikipedia and storing them in a graph structure. These categories are used to group pages by related topics, and are used mostly to find and navigate articles related to a particular subject[1, 2].

Starting from the `Category:Main_topic_classifications` category, the list of sub-categories and documents belonging to that category is recursively

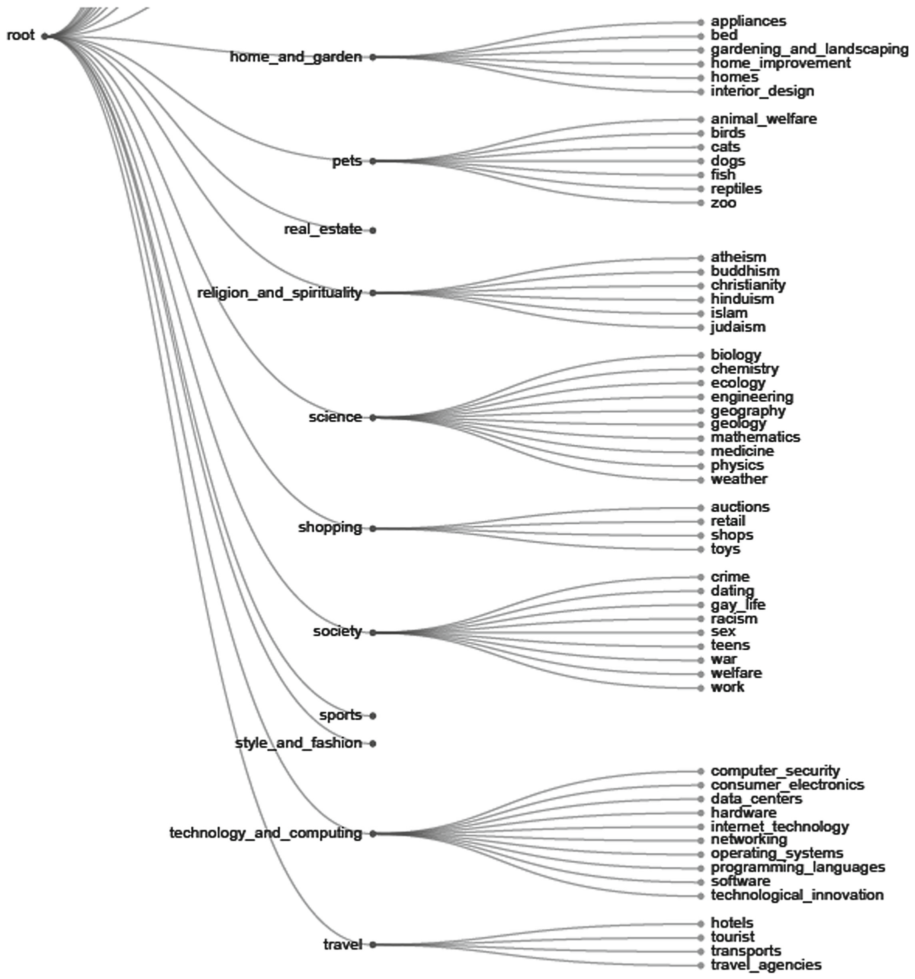


Fig. 2. General-purpose hierarchical taxonomy (lower part).

1 https://www.wikidata.org/wiki/Q2945159.
2 https://en.wikipedia.org/wiki/Help:Category.

retrieved by using the available APIs. This first step results in a graph containing 1.5 million nodes linked together with a subclassing relationship.

The next step of this process involves the computation of the feature vectors, by using the category and sub-category names and, when available, their short description, for each of the nodes in the graph via a word embedding approach. Nodes and vectors are cached locally and the edges are weighted as follows.

Given an edge $e(u, v)$ and the respective vectors V_u and V_v, the weight w_e is:

$$w_e = inverse\_cosine\_similarity(V_u, V_v)$$

This allows NETHIC to have a weighted graph based on the semantic value of Wikipedia's categories, here referred to as *Wikipedia Category Graph*. The subsequent step of the process starts from the *leaf* categories of NETHIC's taxonomy and proceeds by navigating and collecting documents from Wikipedia's categories following the shortest semantic path, until the desired amount of documents are collected. The collected documents are then stored in a structure of folders and sub-folders that follows the structure of NETHIC's taxonomy, where the intermediate folders are formed by using a balanced amount of documents coming from each *leaf* category folder.

3.4 Document Embedding

In the latest years, very important results has been achieved regarding text representation to solve many NLP problems. One of the most renowned solution is a word embedding model, called Word2Vec and proposed by Google [18]. A direct update of this model performs the embedding of sentences or entire documents, and is known as Doc2Vec [15], which is useful to transform sentences or documents into corresponding n-dimensional vectors. This transformation is pivotal since it provides the possibility to work with documents without facing the high dimensionality problem commonly present when a bag-of-words approach is used for text representation. Another advantage for this type of word and document representation lies in the semantic similarity as explained by the authors in their studies. A generic application consists of comparing similar words or document vectors through a cosine similarity metric, in order to evaluate how close two items are in the semantic space. NETHIC uses a Doc2Vec model, trained with an English Wikipedia corpus, following two different strategies. In the first strategy, BOW features are replaced with Doc2Vec vectors, used as features for NETHIC's corpus, in order to verify if sufficiently good results could be obtained with less information. In the second strategy, the BOW functionality is merged with Doc2Vec, so that it may be possible to use the occurrences of words in conjunction with the semantic meaning of the documents.

4 NETHIC's Methodological and Technological Framework

This section reprises and extends NETHIC's framework with respect to the earlier discussion in [6] in terms of its architecture, pre-processing and training mechanisms and algorithms used, underlining the improvements over its earlier version.

4.1 Architecture

The main components of NETHIC's current architecture are artificial neural networks, a hierarchical taxonomy, dictionaries and a Doc2Vec pre-trained model. Figure 3 shows these components and graphically represents the structure and NETHIC's whole process. The latter starts with a text elaboration, by using dictionaries and a document embedding instance to vectorize the input documents, and goes on by relying upon a hierarchical neural networks model to find the main leaf categories to be used as classification labels for the given documents.

4.2 Reasons for a Hierarchical Neural Network Model

In NETHIC, as initially clarified in [6], a neural network hierarchy is employed for several reasons. In fact, in order to classify a document whose main topic is, for instance, *kitchens*, it is sensible to use a neural network that is trained only on texts whose focus is on interior decoration and house supplies, instead of a more heterogeneous or too general artificial neural network.

This avoids the presence of unnecessary words and reduces the noise on the classification process. In NETHIC, for each taxonomy concept, with the exception of the leaf concepts, one neural network is trained and a dictionary of words is built. In the upper levels, the neural networks trained are characterized by a somewhat horizontal view, splitting documents according to general, wide concepts like Economy, Religion, Science and Sports, whereas in the deeper levels the networks tend to assume a more vertical separation and classify the documents according to a more specific category that is a descendant of the general concept (for example, Sports), *e.g. Basketball, Combat Sports, Golf, Soccer, Swimming, Tennis, Volleyball...*

Thus, the classification function used in the neural networks, located at different levels in the hierarchy, is trained with a vocabulary and a set of words, having a varied logical structure and granularity. The upper levels are trained with generic words, which are alike to general "concepts", and the vocabulary used does not include the complete glossary of words associated with the context. When reaching the deeper levels, instead, there is a progressively extensive need to discriminate between semantically-close concepts.

Therefore, the glossary used in the training process contains more specific words, since the classification process, in order to be as effective as possible, has the need to learn additional knowledge on the given area of interest. That is why an iterative approach is followed, particularly suited to artificial neural network-based methods, by descending to the more specific, deeper levels of the classification.

This allows NETHIC to prevent the occurrence of semantic errors when dealing with words belonging to different conceptual areas (like the word *tree*

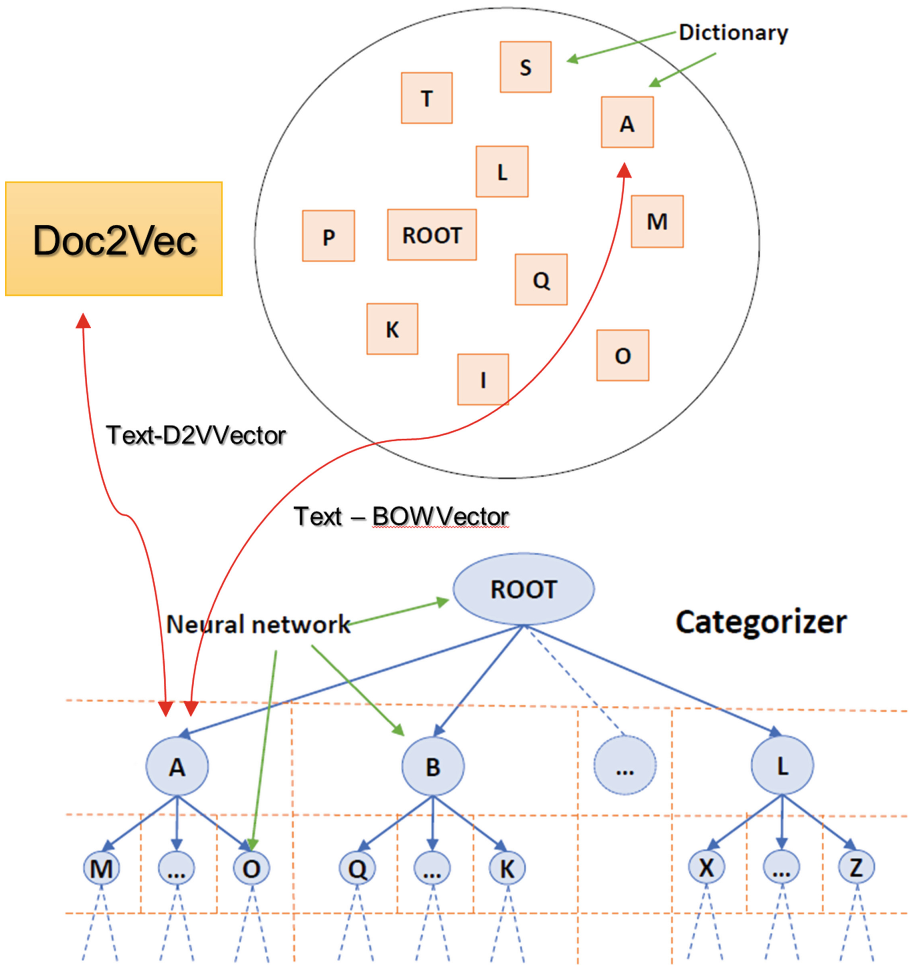


Fig. 3. NETHIC's architecture.

(that represents a plant in the natural world, a component of a ship or a data-representation structure in computer engineering). This sometimes forces the process, when it tries to classify more generic documents, to stop the iteration earlier before it reaches the deepest levels.

In order to understand the potential offered by this approach, let us consider another example where a given document talks about *advertising and marketing*. By taking a look at the taxonomy shown in Figs. 1 and 2, there are many concepts semantically close to the given category such as *personal finance*, *shops* and *movies and tv*, each belonging to different paths. In a scenario where a single neural network is used on 117 different classes, it can be easy to get irrelevant results and low scores due to using the same words in different contexts and with different meanings. In order to face these issues, a hierarchical approach comes thus in handy to decompose the main problem into many sub-classification problems, all of them working together to reduce the noise due to the context by considering trained neural networks on semantically distant concepts.

4.3 Classification Process

The classification process starts with an unstructured text/document as input.

Initially, the *root* category's dictionary and a Doc2Vec (D2V) pre-trained model is used to transform text into a corresponding vectorized form. After that, a BOW+Doc2Vec composed vector is passed as input to *root's Neural Network* to perform the prediction task. As explained in Sect. 4.6, the first relevant categories are chosen to continue with the next steps, considering appropriate dictionaries and neural networks.

4.4 Data Pre-processing

The pre-processing step is required to transform the unstructured datasets explained in Sect. 3.3 in order to obtain a useful and structured version of the data. Before delving deeper into the pre-processing step, it is noteworthy to say that the initial corpus has been split into two balanced corpora with a ratio of 95% - 5%. The first corpus, named *Corpus-A* and containing 54.439 documents (about 465 for each leaf category), has been used for the training and validation tasks on single neural networks. The second corpus, called *Corpus-B*, containing 2843 documents (about 25 for each leaf category), has been used throughout the entire validation of the hierarchical model. As known in literature as well as in commercial environments, an ETL (Extraction, Transformation and Loading) process represents a key point for data collection and feature extraction tasks. In this work, three kinds of transformations, explained below, are used in order to build a sufficient number of datasets to check and identify the best features to be used.

The details of the first transformation can be found in [6]; it produces BOW-based datasets that will be referred to as *Datasets_BOW* from now on. Dictionaries used in the hierarchical validation step are saved in order to transform the validation corpus by considering the same words already used to train the neural networks.

A second transformation used the Doc2Vec model to convert documents into suitable vectors of 300 dimensions. Unlike the first transformation, the built datasets called *Datasets_D2V* consume a really slight portion of memory and there is no need to store dictionaries for the subsequent validation phase.

Finally, the two abovementioned transformations in order to use both features type. In this case the datasets obtained called *Datasets_D2V-BOW* are far too large to be kept in memory, just like the BOW-based datasets. The resulting vectors will be in this case the concatenation of the BOW and D2V vectors, thus dictionaries are saved here as well.

For each of these transformations, 18 datasets are built.

The next subsection describes how three corresponding models of neural networks, each for one of the three transformations, have been trained and compared.

4.5 Training

As mentioned in the previous subsection, the training phase carried out by using *Corpus-A* has been performed three times, one for each transformation (and thus for each group of datasets). In this subsection, firstly the generic method used to train the single neural networks will be explained, and then the different models will be compared in order to find the best features to be used for the classification problem.

According to the best practices for training artificial intelligence models, a cross-validation was executed to check for potential overfitting/underfitting, by using the k-fold and "leave one out approach" [30]. By resorting to this technique, an initial, balanced splitting of the datasets has been necessary to compute the training and testing in "leave one out". Starting from this assumption, for any single dataset two sub-datasets have been built, with 90% and 10% proportions, respectively. For example, considering a theoretical category *X* and a corresponding BOW dataset saved as *Dataset_BOW_X*, a splitting is made in order to obtain *Dataset_BOW_X_Training_CV* and *Dataset_BOW_X_Validation_LOO*. The following pseudocode describes how the training phase was performed.

Algorithm 1. Training using Cross-Validation and Test in One Shot.

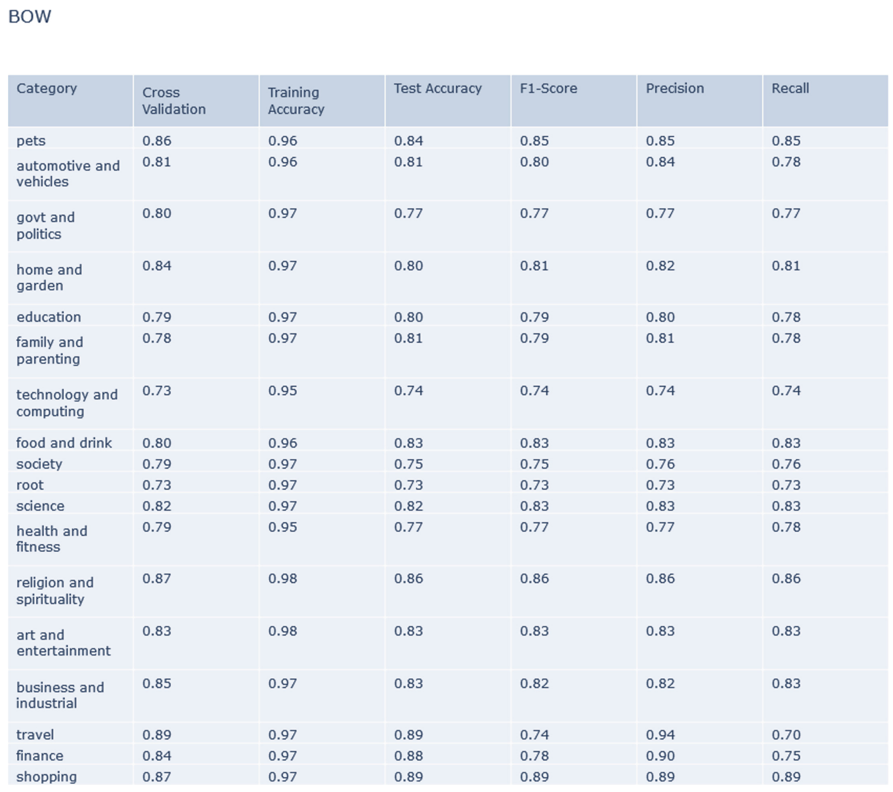
```
 1: procedure TRAINING
 2:     for middle Taxonomy's Category X do
 3:         Dataset_X_Training_CV ←Dataset_X
 4:         Dataset_X_Validation_LOO ←Dataset_X
 5:         CV_accuracy ←0
 6:         for each folds combinations (4,1) from Dataset_X_Training_CV do
 7:             current_CV_model_X ←training(4_ folds)
 8:             current_CV_accuracy_X ←model.validation(1_fold)
 9:             CV_accuracy ←CV_accuracy + current_CV_accuracy_X
10:         CV_accuracy ←CV_accuracy : 5
11:         model_X ←training(Dataset_X_Training_CV)
12:         model_accuracy_X ←model.validation(Dataset_X_Validation_LOO)
13:         save(model_X)
```

Basically, any category used to realize the hierarchical model covers all the steps described in Algorithm 1, and for each of them, a cross-validation has been performed in order to evaluate the potential presence of underfitting and overfitting. After making sure that none of these problems had arisen, it was possible to train the neural network using *Dataset_X_Training_CV* subsequently validated with *Dataset_X_Validation_LOO*. This algorithm has been executed on the three groups of datasets previously described, obtaining neural networks for each of the features considered, that is to say BOW, Doc2Vec and BOW-Doc2Vec. The three tables showed in Figs. 4, 5 and 6, respectively, contain Cross-Validation Accuracy, Training Accuracy, Test Accuracy, Precision, Recall and F1-score metrics for each of the trained models. As shown, the best accuracies are obtained with the combined model that uses BOW and D2V features together. The worst performance was obtained by using the model trained on D2V features only: this means that for this kind of complex classification, document embedding by itself is not a good choice to represent documents, but it can nevertheless be useful to improve the accuracy of the BOW model, as seen in the results obtained. By considering the BOW and BOW-D2V accuracy values, there is an improvement of about 2% for most categories, and an improvement of about 1% for the *root* category: this is especially important, because in the hierarchical model it represents the heaviest category for the correct construction of the classification paths. Cross-validation results show that there are no overfitting and underfitting issues exactly as expected. Training and test accuracies show that all the trained models learn well and are able to generalize with data never seen before. Precision, Recall and F1-Score show that the trained models are able to obtain a good accuracy for all of the labels, and in general they do not confuse among classes that are semantically close to one another.

4.6 Algorithm to Build Paths

The algorithm used to build paths has not undergone significant modifications from the one described in [6]. For each returned path, the average between all the single scores for each of the corresponding categories is computed.

For instance, given the following path: $\mathtt{P} = \mathtt{C}_1/\mathtt{C}_2/C_3$ with its respective scores $\mathtt{SC}_1$, $\mathtt{SC}_2$ and $\mathtt{SC}_2$, its corresponding total score $\mathtt{S}_\mathtt{P}$ is the average of the single scores. The system keeps considering categories until the probabilities returned by the current neural networks reach a threshold that is initially set as `0.7`. If after the first classification a good current tolerance is obtained, this will consequently lead to a reasonable classification; otherwise, if such a value is low, it means that there are paths with a low score. In this case, a second classification iteration is run by considering the paths with a lower score value. In general, this algorithm allows the system to select the highest-level categories and concepts when the textual content examined contains only generic terms, whereas it is possible to select more detailed and low-level categories and concepts by examining texts that are very specific, technical or focused on a certain topic.



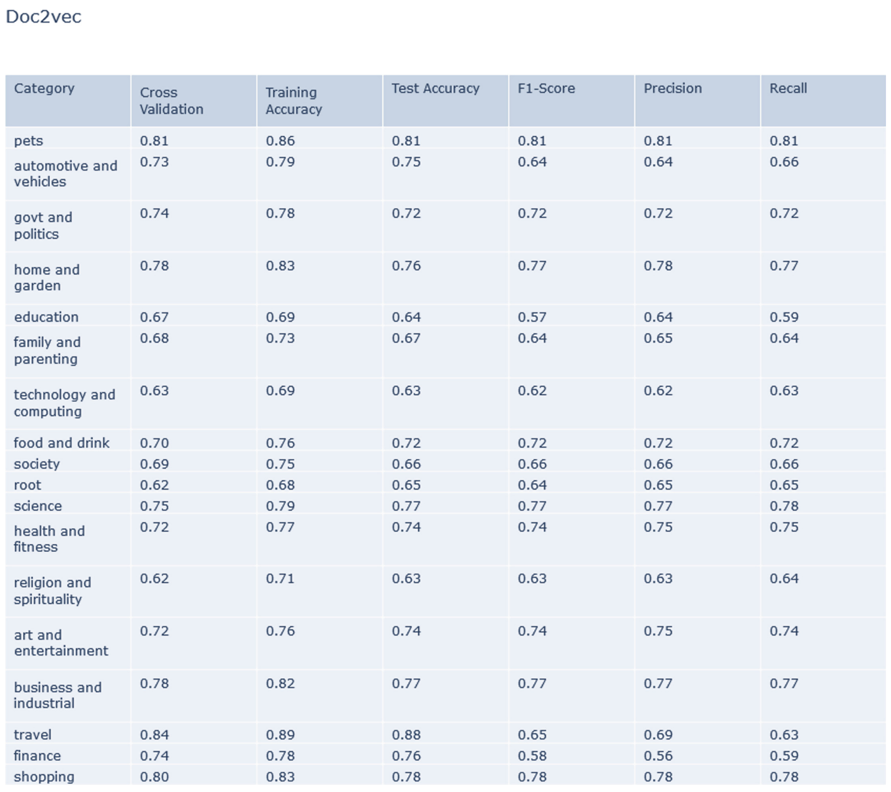
BOW

Category	Cross Validation	Training Accuracy	Test Accuracy	F1-Score	Precision	Recall
pets	0.86	0.96	0.84	0.85	0.85	0.85
automotive and vehicles	0.81	0.96	0.81	0.80	0.84	0.78
govt and politics	0.80	0.97	0.77	0.77	0.77	0.77
home and garden	0.84	0.97	0.80	0.81	0.82	0.81
education	0.79	0.97	0.80	0.79	0.80	0.78
family and parenting	0.78	0.97	0.81	0.79	0.81	0.78
technology and computing	0.73	0.95	0.74	0.74	0.74	0.74
food and drink	0.80	0.96	0.83	0.83	0.83	0.83
society	0.79	0.97	0.75	0.75	0.76	0.76
root	0.73	0.97	0.73	0.73	0.73	0.73
science	0.82	0.97	0.82	0.83	0.83	0.83
health and fitness	0.79	0.95	0.77	0.77	0.77	0.78
religion and spirituality	0.87	0.98	0.86	0.86	0.86	0.86
art and entertainment	0.83	0.98	0.83	0.83	0.83	0.83
business and industrial	0.85	0.97	0.83	0.82	0.82	0.83
travel	0.89	0.97	0.89	0.74	0.94	0.70
finance	0.84	0.97	0.88	0.78	0.90	0.75
shopping	0.87	0.97	0.89	0.89	0.89	0.89

Fig. 4. Single ANN's scores with BOW dataset.

5 Experimentation of NETHIC's Extended Method and Comparison with the Earlier Method

In this section the results of the new experimentation carried out after the introduction of the combined BOW+Doc2Vec document embedding mechanism is reported and compared with the earlier version of NETHIC (with only the BOW mechanism). The focus here is on the pie charts and confusion matrices that show how integrating the Doc2Vec model for feature extraction is a sound approach combined with the earlier BOW-based method. For the purposes of such a comparison, the terms "NETHIC" and "NETHIC-2" will be used to differentiate between NETHIC's original approach and the extended one, respectively. The last part of this section discusses a couple of practical examples to conclude the analysis.



Doc2vec

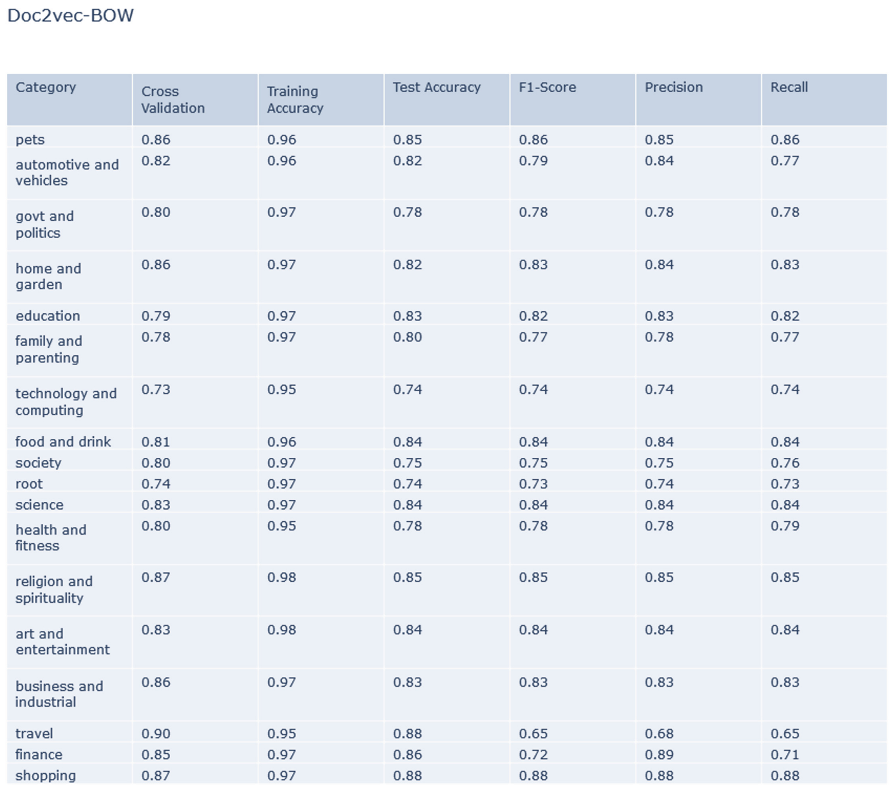
Category	Cross Validation	Training Accuracy	Test Accuracy	F1-Score	Precision	Recall
pets	0.81	0.86	0.81	0.81	0.81	0.81
automotive and vehicles	0.73	0.79	0.75	0.64	0.64	0.66
govt and politics	0.74	0.78	0.72	0.72	0.72	0.72
home and garden	0.78	0.83	0.76	0.77	0.78	0.77
education	0.67	0.69	0.64	0.57	0.64	0.59
family and parenting	0.68	0.73	0.67	0.64	0.65	0.64
technology and computing	0.63	0.69	0.63	0.62	0.62	0.63
food and drink	0.70	0.76	0.72	0.72	0.72	0.72
society	0.69	0.75	0.66	0.66	0.66	0.66
root	0.62	0.68	0.65	0.64	0.65	0.65
science	0.75	0.79	0.77	0.77	0.77	0.78
health and fitness	0.72	0.77	0.74	0.74	0.75	0.75
religion and spirituality	0.62	0.71	0.63	0.63	0.63	0.64
art and entertainment	0.72	0.76	0.74	0.74	0.75	0.74
business and industrial	0.78	0.82	0.77	0.77	0.77	0.77
travel	0.84	0.89	0.88	0.65	0.69	0.63
finance	0.74	0.78	0.76	0.58	0.56	0.59
shopping	0.80	0.83	0.78	0.78	0.78	0.78

Fig. 5. Single ANN's scores with Doc2Vec dataset.

5.1 Comparison Between NETHIC and NETHIC-2

As explained in [6], to evaluate the tool's accuracy the first three categories returned by the algorithm to build paths detailed in Sect. 4.6 are considered. This choice is meaningful because when many classes—some of them semantically close to one another—are used for text classification, a single assigned class may not be the only and optimal solution. For this comparison *Corpus_B*, which contains 2843 documents (about 25 for each leaf category), has been used to test both methods.

Clearly, dictionaries are used step-by-step for each different path in order to build the correct BOW vector to be merged with the unchanged Doc2Vec vector (which stays the same for every document to be classified), in order to keep the coherence with the currently analyzed category. The pie charts in Fig. 7 emphasize the improvement obtained with the extended method, which is able to correctly classify ∼60 documents more than the earlier approach. The improvement observed during the training phase is the same as in this evaluation, and therefore confirms an overall improvement of 2%.



Doc2vec-BOW

Category	Cross Validation	Training Accuracy	Test Accuracy	F1-Score	Precision	Recall
pets	0.86	0.96	0.85	0.86	0.85	0.86
automotive and vehicles	0.82	0.96	0.82	0.79	0.84	0.77
govt and politics	0.80	0.97	0.78	0.78	0.78	0.78
home and garden	0.86	0.97	0.82	0.83	0.84	0.83
education	0.79	0.97	0.83	0.82	0.83	0.82
family and parenting	0.78	0.97	0.80	0.77	0.78	0.77
technology and computing	0.73	0.95	0.74	0.74	0.74	0.74
food and drink	0.81	0.96	0.84	0.84	0.84	0.84
society	0.80	0.97	0.75	0.75	0.75	0.76
root	0.74	0.97	0.74	0.73	0.74	0.73
science	0.83	0.97	0.84	0.84	0.84	0.84
health and fitness	0.80	0.95	0.78	0.78	0.78	0.79
religion and spirituality	0.87	0.98	0.85	0.85	0.85	0.85
art and entertainment	0.83	0.98	0.84	0.84	0.84	0.84
business and industrial	0.86	0.97	0.83	0.83	0.83	0.83
travel	0.90	0.95	0.88	0.65	0.68	0.65
finance	0.85	0.97	0.86	0.72	0.89	0.71
shopping	0.87	0.97	0.88	0.88	0.88	0.88

Fig. 6. Single ANN's scores with BOW+Doc2Vec dataset.

The following confusion matrix shows the methods' accuracy for the first hierarchical level in order to understand the improvement for the root neural network. The diagonal values for the matrix in both Figs. 8 and 9 represents the correct classifications and make the matrix almost diagonal. The best performance for the *Science* category is obtained by NETHIC-2 with about 8 more documents that with the earlier method had been lost. In general, improvements over NETHIC's previous method can be seen in *Art and Entertainment*, which is now less confused with other categories, in *Society*, which is now less confused with *Family_and_parenting*, and in *Health_and_fitness*, previously more confused with a lot of other categories containing similar contents like *Society, Sport and Food_and_drink*. In computational terms there are no relevant differences, since the addition of a 300-sized Doc2Vec vector does not change the order of magnitude of the feature vectors to be used for the training and classification steps.

5.2 Examples

Last but not least, practical classification examples are reported by showing two different Wikipedia documents. In the first example, a document that talks

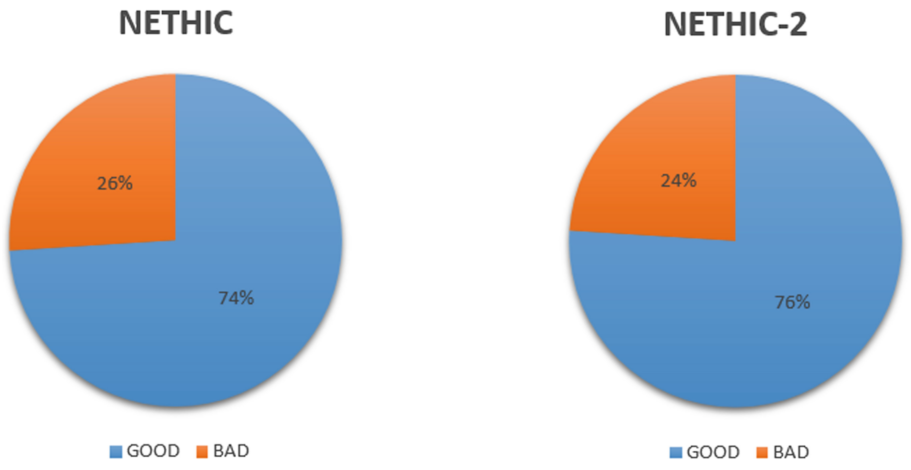


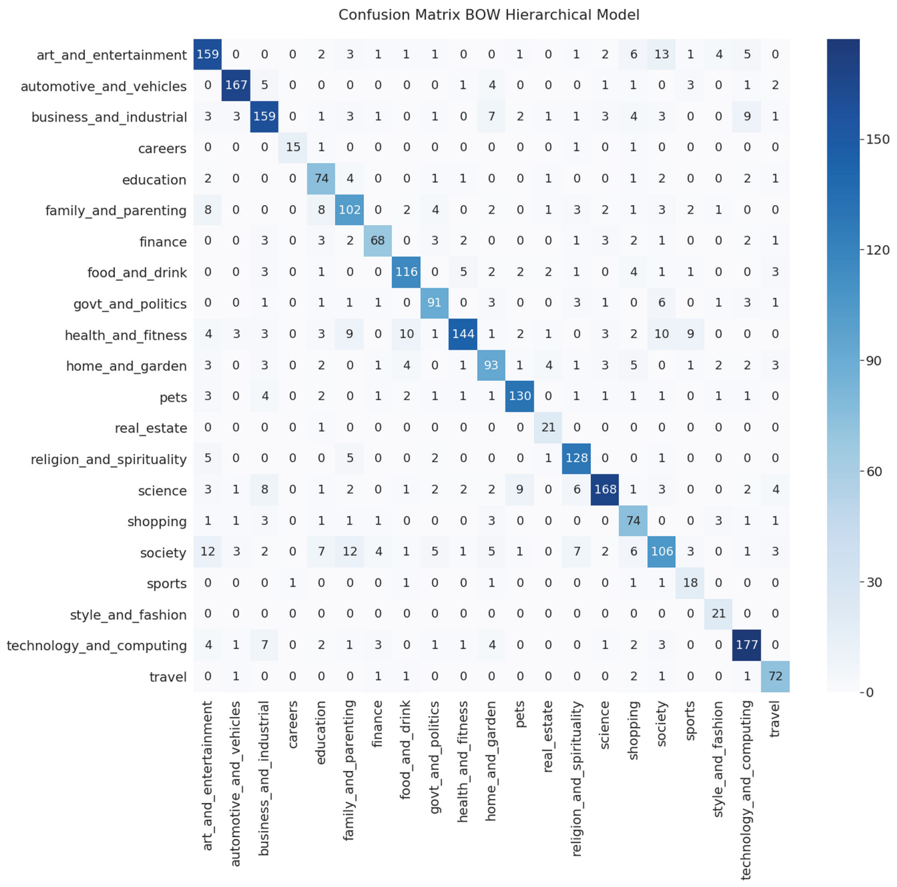
Fig. 7. Classification accuracy of the initial (leftmost chart) and the extended method (rightmost chart).

about a specific mineral called "Bukovskyite", and was labeled in *Corpus_B* as *Iron_and_steel_industry*, has been classified correctly as *business_and_industrial-/iron_and_steel_industry/*, as well as *science/geology/* that is correct for obvious reasons. In the second example a document talking about food-related problems has been classified. As shown, the classifier returned categories including the correct label *food_and_drink/healthy_eating/* as the second choice, which may be considered a good result, but also contains a more relevant category for such a document like *health_and_fitness/addiction/*, which constitutes a surprising achievement.

Iron_and_steel_industry Wikipedia Document. *Bukovskyite (also known as "clay of Kutná Hora") is an iron arsenate sulfate mineral which forms nodules with a reniform (kidney-shaped) surface. Under a microscope, these nodules appear as a collection of minute needles similar to gypsum. Some can be seen with the naked eye and occur inside the nodules. Bukovskyite was first described from pit heaps from the Middle Ages, where sulfate ores had been mined at Kank, north of Kutná Hora in Bohemia, Czech Republic, and other old deposits in the vicinity. Only recently defined and acknowledged, it was approved by the IMA in 1969. Bukovskyite was collected a long time ago from the overgrown pit heaps by the inhabitants of Kutná Hora. It was used for poisoning field mice and other field vermin. This poisonous clay, known also by the place name as "clay of Kutná Hora", was widely known and it was considered to be arsenic (arsenic trioxide).*

Classification Results

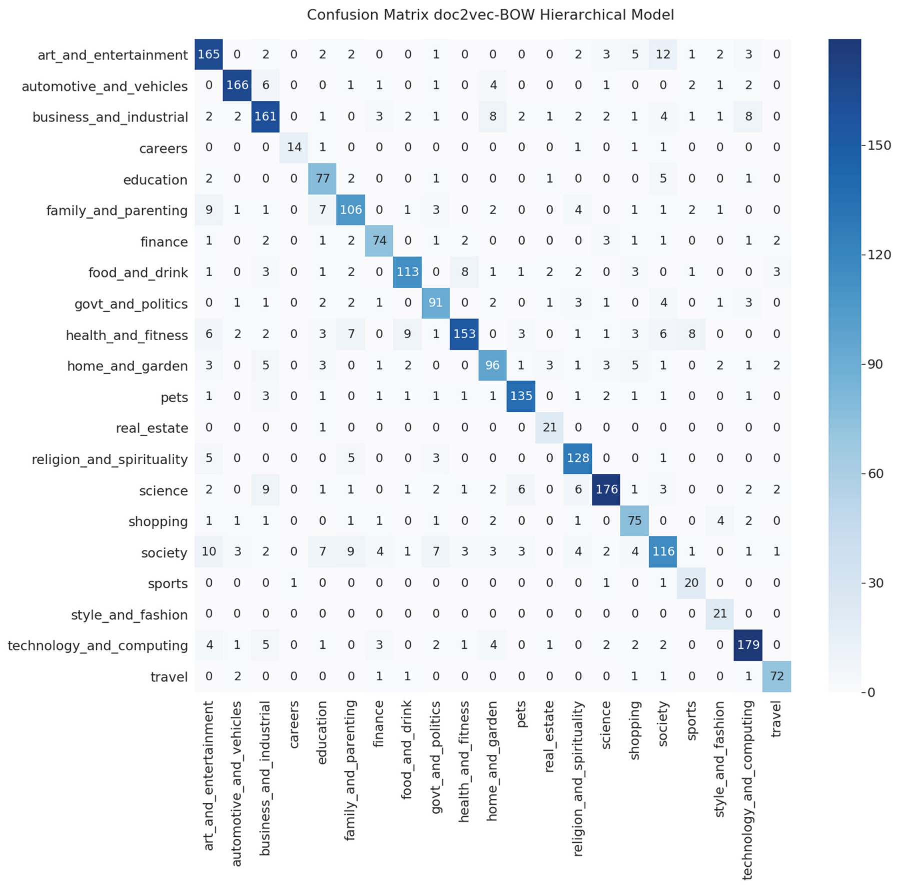
- **Label** = *business_and_industrial/iron_and_steel_industry/* **Score** = 0.68
- **Label** = *science/geology/* **Score** = 0.53



	art_and_entertainment	automotive_and_vehicles	business_and_industrial	careers	education	family_and_parenting	finance	food_and_drink	govt_and_politics	health_and_fitness	home_and_garden	pets	real_estate	religion_and_spirituality	science	shopping	society	sports	style_and_fashion	technology_and_computing	travel
art_and_entertainment	159	0	0	0	2	3	1	1	1	0	0	1	0	1	2	6	13	1	4	5	0
automotive_and_vehicles	0	167	5	0	0	0	0	0	0	1	4	0	0	0	1	1	0	3	0	1	2
business_and_industrial	3	3	159	0	1	3	1	0	1	0	7	2	1	1	3	4	3	0	0	9	1
careers	0	0	0	15	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
education	2	0	0	0	74	4	0	0	1	1	0	0	1	0	0	1	2	0	0	2	1
family_and_parenting	8	0	0	0	8	102	0	2	4	0	2	0	1	3	2	1	3	2	1	0	0
finance	0	0	3	0	3	2	68	0	3	2	0	0	0	1	3	2	1	0	0	2	1
food_and_drink	0	0	3	0	1	0	0	116	0	5	2	2	2	1	0	4	1	1	0	0	3
govt_and_politics	0	0	1	0	1	1	1	0	91	0	3	0	0	3	1	0	6	0	1	3	1
health_and_fitness	4	3	3	0	3	9	0	10	1	144	1	2	1	0	3	2	10	9	0	0	0
home_and_garden	3	0	3	0	2	0	1	4	0	1	93	1	4	1	3	5	0	1	2	2	3
pets	3	0	4	0	2	0	1	2	1	1	1	130	0	1	1	1	1	0	1	1	0
real_estate	0	0	0	0	1	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0
religion_and_spirituality	5	0	0	0	0	5	0	0	2	0	0	0	1	128	0	0	1	0	0	0	0
science	3	1	8	0	1	2	0	1	2	2	2	9	0	6	168	1	3	0	0	2	4
shopping	1	1	3	0	1	1	1	0	0	0	3	0	0	0	0	74	0	0	3	1	1
society	12	3	2	0	7	12	4	1	5	1	5	1	0	7	2	6	106	3	0	1	3
sports	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	1	18	0	0	0
style_and_fashion	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0
technology_and_computing	4	1	7	0	2	1	3	0	1	1	4	0	0	0	1	2	3	0	0	177	0
travel	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	2	1	0	0	1	72

Fig. 8. NETHIC's original results (with only the BOW-based embedding mechanism).

Healthy Eating Wikipedia Document. *Overeaters Anonymous (OA) is a twelve-step program for people with problems related to food including, but not limited to, compulsive overeaters, those with binge eating disorder, bulimics and anorexics. Anyone with a problematic relationship with food is welcomed, as OA's Third Tradition states that the only requirement for memberships is a desire to stop eating compulsively. OA was founded by Rozanne S. and two other women in January 1960. The organizationś headquarters, or World Service Office, is located in Rio Rancho, New Mexico. Overeaters Anonymous estimates its membership at over 60,000 people in about 6,500 groups meeting in over 75 countries. OA has developed its own literature specifically for those who eat compulsively but also uses the Alcoholics Anonymous books Alcoholics Anonymous and Twelve Steps and Twelve Traditions. The First Step of OA begins with the admission of powerlessness over food; the next eleven steps are intended to bring members physical, emotional, and spiritual healing.*



	art_and_entertainment	automotive_and_vehicles	business_and_industrial	careers	education	family_and_parenting	finance	food_and_drink	govt_and_politics	health_and_fitness	home_and_garden	pets	real_estate	religion_and_spirituality	science	shopping	society	sports	style_and_fashion	technology_and_computing	travel
art_and_entertainment	165	0	2	0	2	2	0	0	1	0	0	0	0	2	3	5	12	1	2	3	0
automotive_and_vehicles	0	166	6	0	0	1	1	0	1	0	4	0	0	0	1	0	0	2	1	2	0
business_and_industrial	2	2	161	0	1	0	3	2	1	0	8	2	1	2	2	1	4	1	1	8	0
careers	0	0	0	14	1	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0
education	2	0	0	0	77	2	0	0	1	0	0	0	1	0	0	0	5	0	0	1	0
family_and_parenting	9	1	1	0	7	106	0	1	3	0	2	0	0	4	0	1	1	2	1	0	0
finance	1	0	2	0	1	2	74	0	1	2	0	0	0	0	3	1	1	0	0	1	2
food_and_drink	1	0	3	0	1	2	0	113	0	8	1	1	2	2	0	3	0	1	0	0	3
govt_and_politics	0	1	1	0	2	2	1	0	91	0	2	0	1	3	1	0	4	0	1	3	0
health_and_fitness	6	2	2	0	3	7	0	9	1	153	0	3	0	1	1	3	6	8	0	0	0
home_and_garden	3	0	5	0	3	0	1	2	0	0	96	1	3	1	3	5	1	0	2	1	2
pets	1	0	3	0	1	0	1	1	1	1	1	135	0	1	2	1	1	0	0	1	0
real_estate	0	0	0	0	1	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0
religion_and_spirituality	5	0	0	0	0	5	0	0	3	0	0	0	0	128	0	0	1	0	0	0	0
science	2	0	9	0	1	1	0	1	2	1	2	6	0	6	176	1	3	0	0	2	2
shopping	1	1	1	0	0	1	1	0	1	0	2	0	0	1	0	75	0	0	4	2	0
society	10	3	2	0	7	9	4	1	7	3	3	3	0	4	2	4	116	1	0	1	1
sports	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	20	0	0	0
style_and_fashion	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0
technology_and_computing	4	1	5	0	1	0	3	0	2	1	4	0	1	0	2	2	2	0	0	179	0
travel	0	2	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	1	72

Fig. 9. NETHIC's results with the introduction of the combined BOW+Doc2Vec embedding mechanism.

Classification Results

- **Label** = *health_and_fitness/addiction/* **Score** = 0.64
- **Label** = *food_and_drink/healthy_eating/* **Score** = 0.38
- **Label** = *food_and_drink/gastronomy/* **Score** = 0.26

5.3 Technical Configuration for the Experimentation

The hardware configuration employed for the reported experimentation includes the following systems: one Intel i7-6700HQ CPU with 16 GB DDR3 RAM, one Intel i7-7700 CPU with 32 GB DDR3 RAM and Sandisk Ultra SSD, and one Intel i7-8550U CPU with 32 GB DDR4 RAM and Samsung Pro SSD. The classifier used in NETHIC has been written in Python and exploits the *scikit-learn*, *CountVectorizer* and *Multi-layer Perceptron (MLP)* libraries to create the feature vectors and the artificial neural networks themselves. For the Doc2Vec pretrained model, the *Gensim* library has been used. Persistence and loading of the networks is done via the *pickle* library.

6 Discussion and Conclusion

This work reported and extended the discussion on NETHIC, a software tool implementing a classification method for textual documents relying upon hierarchical taxonomies, artificial neural networks and a document embedding mechanism.

The earlier research discussed in [6] proved the combination of artificial neural networks and hierarchical taxonomies to be effective for tackling the classification problem, displaying an overall solid performance together with relevant characteristics of scalability and modularity.

With respect to the initial version of NETHIC, the current tool now takes advantage of a state-of-art Natural Language Processing technique like Doc2Vec, and the results achieved with the introduction of such an embedding technique, in combination with the earlier used bag-of-words, have demonstrated that with a slight increase in the dimensional space it is possible to obtain better results in the classification of documents and texts.

In this regard, the experimentation reported in this work showed that the improvements obtained with respect of NETHIC's original method via the combination of the BOW and Doc2Vec embedding mechanisms encourages their combined usage so that more information can be considered by NETHIC's neural networks in order for it to understand and choose the correct categories for classification. Taken individually, the BOW mechanism proved to be sufficiently solid (as seen in [6]), whereas it may not be advisable to use Doc2Vec by itself, probably because semantically-close categories, like the leaves of a given intermediate category, are difficult to be told apart without considering the words used.

Future work may explore the possibilities of integrating and/or extending other state-of-the art methods like BERT [8] that are currently heading towards ever newer and future-envisioning frontiers.

References

1. Atzeni, P., Polticelli, F., Toti, D.: An automatic identification and resolution system for protein-related abbreviations in scientific papers. In: Pizzuti, C., Ritchie, M.D., Giacobini, M. (eds.) EvoBIO 2011. LNCS, vol. 6623, pp. 171–176. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20389-3_18
2. Atzeni, P., Polticelli, F., Toti, D.: Experimentation of an automatic resolution method for protein abbreviations in full-text papers. In: 2011 ACM Conference on Bioinformatics, Computational Biology and Biomedicine, BCB 2011, pp. 465–467 (2011). https://doi.org/10.1145/2147805.2147871

3. Atzeni, P., Polticelli, F., Toti, D.: A framework for semi-automatic identification, disambiguation and storage of protein-related abbreviations in scientific literature. In: Proceedings - International Conference on Data Engineering, pp. 59–61 (2011). https://doi.org/10.1109/ICDEW.2011.5767646
4. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python - Analyzing Text with the Natural Language Toolkit. O'Reilly, Sebastopol (2009)
5. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. CoRR abs/1710.05381 (2017). http://arxiv.org/abs/1710.05381
6. Ciapetti, A., Florio, R.D., Lomasto, L., Miscione, G., Ruggiero, G., Toti, D.: NETHIC: a system for automatic text classification using neural networks and hierarchical taxonomies. In: ICEIS 2019 - Proceedings of the 21st International Conference on Enterprise Information Systems, pp. 284–294 (2019). https://doi.org/10.5220/0007709702960306
7. Dalal, M.K., Zaveri, M.: Automatic text classification: a technical review. Int. J. Comput. Appl. **28** (2011)
8. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018). http://arxiv.org/abs/1810.04805
9. Forman, G.: An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. **3**, 1289–1305 (2003)
10. Ha, J.W., Pyo, H., Kim, J.: Large-scale item categorization in e-commerce using multiple recurrent neural networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 107–115. ACM, New York (2016). https://doi.org/10.1145/2939672.2939678
11. Hermundstad, A., Brown, K., Bassett, D., Carlson, J.: Learning, memory, and the role of neural network architecture. PLoS Comput. Biol. **7**, e1002063 (2011)
12. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 655–665. Association for Computational Linguistics, Baltimore, June 2014. http://www.aclweb.org/anthology/P14-1062
13. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, 25–29 October 2014, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1746–1751 (2014). http://aclweb.org/anthology/D/D14/D14-1181.pdf
14. Koppel, M., Winter, Y.: Determining if two documents are written by the same author. J. Assoc. Inf. Sci. Technol. **65**, 178–187 (2014)
15. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML 2014, pp. II-1188–II-1196 (2014). http://dl.acm.org/citation.cfm?id=3044805.3045025. JMLR.org
16. Lewis, D.D., Ringuette, M.: A comparison of two learning algorithms for text categorization. In: Third Annual Symposium on Document Analysis and Information Retrieval, pp. 81–93 (1994)
17. McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: Learning for Text Categorization: Papers from the 1998 AAAI Workshop, pp. 41–48 (1998). http://www.kamalnigam.com/papers/multinomial-aaaiws98.pdf

18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS 2013, pp. 3111–3119. Curran Associates Inc. (2013). http://dl.acm.org/citation.cfm?id=2999792.2999959
19. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**, 1–47 (2002)
20. Shen, D., Ruvini, J.D., Mukherjee, R., Sundaresan, N.: A study of smoothing algorithms for item categorization on e-commerce sites. Neurocomputing **92**, 54–60 (2012). https://doi.org/10.1016/j.neucom.2011.08.035
21. Silla Jr., C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Min. Knowl. Discov. **22**(1–2), 31–72 (2011). https://doi.org/10.1007/s10618-010-0175-9
22. Toti, D., Atzeni, P., Polticelli, F.: Automatic protein abbreviations discovery and resolution from full-text scientific papers: the PRAISED framework. Bio Algorithms Med Syst. **8** (2012). https://doi.org/10.2478/bams-2012-0002
23. Toti, D., Rinelli, M.: On the road to speed-reading and fast learning with CONCEPTUM. In: Proceedings - 2016 International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS 2016, pp. 357–361 (2016). https://doi.org/10.1109/INCoS.2016.30
24. Vidhya, K., Aghila, G.: A survey of Naive Bayes machine learning approach in text document classification. Int. J. Comput. Sci. Inf. Secur. **7**, 206–211 (2010)
25. W3C: Skos - simple knowledge organization system reference (2009). https://www.w3.org/TR/2009/REC-skos-reference-20090818/
26. W3C: RDF resource description framework (2014). http://www.w3.org/RDF/
27. Wang, L., Zhao, X.: Improved K-NN classification algorithm research in text categorization. In: Proceedings of the 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 1848–1852 (2012)
28. Wang, S., Manning, C.: Baselines and bigrams: simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the ACL: Short Papers, vol. 2, pp. 90–94. ACL (2012)
29. Wetzker, R., et al.: Tailoring taxonomies for efficient text categorization and expert finding. In: 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 459–462, December 2008
30. Wong, T.T.: Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern Recogn. **48**(9), 2839–2846 (2015). https://doi.org/10.1016/j.patcog.2015.03.009
31. Zhang, Y., Roller, S., Wallace, B.C.: MGNC-CNN: a simple approach to exploiting multiple word embeddings for sentence classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1522–1527. Association for Computational Linguistics, San Diego, June 2016. https://doi.org/10.18653/v1/N16-1178
32. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 253–263. Asian Federation of Natural Language Processing, Taipei, November 2017. https://www.aclweb.org/anthology/I17-1026
33. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Trans. Knowl. Data Eng. **18**(1), 63–77 (2006). https://doi.org/10.1109/TKDE.2006.17