A Domain Categorisation of Vocabularies Based on a Deep Learning Classifier

Alberto Nogales[a,*], Álvaro García-Tejedor[a] and Miguel-Angel Sicilia[b]
[a]*CEIEC, Research Institute, Francisco de Vitoria University, Ctra. M-515 Pozuelo-Majadahonda km. 1,800, 28223 Pozuelo de Alarcón, Spain*
[b]*Information Engineering Research Unit, Computer Science Department, University of Alcalá de Henares, Ctra. Barcelona km. 33.6, 28871 Alcalá de Henares, Spain*

Abstract. The publication of large amounts of open data has become a major trend nowadays. This is a consequence of projects like the Linked Open Data (LOD) community, which publishes and integrates datasets using techniques like Linked Data. Linked Data publishers should follow a set of principles for dataset design. This information is described in a 2011 document that describes tasks as the consideration of reusing vocabularies. With regard to the latter, another project called Linked Open Vocabularies (LOV) attempts to compile the vocabularies used in LOD. These vocabularies have been classified by domain following the subjective criteria of LOV members, which has the inherent risk introducing personal biases. In this paper, we present an automatic classifier of vocabularies based on the main categories of the well-known knowledge source Wikipedia. For this purpose, word-embedding models were used, in combination with Deep Learning techniques. Results show that with a hybrid model of regular Deep Neural Network (DNN), Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN), vocabularies could be classified with an accuracy of 93.57 per cent. Specifically, 36.25 per cent of the vocabularies belong to the Culture category.

Keywords: Linked Data, Deep Learning, Document Categorisation.

1. Introduction

In recent years, the Linked Data technique has emerged for publishing and integrating structured data. In order to achieve data standardisation, data providers should follow the Linked Data principles formulated by Tim Berners Lee in 2006 [1]. Its use brought the appearance of projects like the Linked Open Data community (LOD), which aim to publish and interlink open datasets [2]. This is achieved by using Resource Description Framework (RDF) [3] to describe the data, and RDF links to interlink the datasets. The objective is to build a global space, called the Web of Linked Data, by reusing data sources. A graphical representation of its structure can be seen in the LOD cloud [4].

Another important document is [5], which compiles the best practices in Linked Data and which reviews the reuse of vocabularies. Also, in a document called Best Practices for Publishing Linked Data from 2014, it is recommended that vocabularies are reused whenever is possible [6]. As a consequence of these recommendations, the Linked Open Vocabularies (LOV) project was developed [7]. This aims to compile the vocabularies used by LOD, ensuring that they are easy to access, and providing general metrics and statistics regarding their characteristics. These vocabularies are domain classified with a set of tags for ease of use.

In contrast to LOV, publishers in LOD are responsible for the domain categorisation of datasets. Hence, domain categorisation in LOV is based solely on the personal criteria of its members. As those who create the vocabulary are not the same individuals that decide their scope in LOV, this risks a biased classification. It should also be noted that this is a tedious and

*Corresponding author. E-mail: alberto.nogales@ceiec.es

time-consuming task. In order to make this process easier, and to avoid the use of personal criteria, it makes sense to benefit from using Deep Learning techniques. These models are useful for obtaining patterns in high dimensional datasets in order to classify new instances.

To give an example of vocabulary classified by an individual, terms relating to videogames might be classified as 'Culture' or 'Technology' depending on the background of the person who is making the decision. Deep Learning techniques are based on neural network models and are increasingly used in the field of machine learning. Deep Learning is defined by [8] as a technique that uses computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. In other words, machines learn patterns or structures in large sets of data by adjusting the parameters of neural networks with more than one layer. By applying Deep Learning, vocabularies in LOV could be categorised automatically using objective criteria based on their similarity.

This paper considers the field of document classification or document categorisation. In particular, it will be adapted to the vocabularies of LOV. To this end, the study will refer to ontology (as these vocabularies are ontologies) classification/categorisation. The experiment is structured in two stages, each using different elements: the dump provided by LOV with all the vocabularies, and the creation of a corpus based on Wikipedia's[1] Main categories and the Deep Learning model for classifying. The first provided a set of texts tagged with a domain. The corpus was used to train the model to make accurate classifications. Finally, the model was used to automatically categorise the vocabularies.

Results show that the model obtained can classify vocabularies with an accuracy of 93 per cent. The most commonly used categories are Culture (which is the largest, with 36.25 per cent of the vocabularies), followed by Nature, Society, History and People, which each have around 10 per cent.

The rest of this paper is structured as follows. Section 2 provides a brief overview of the current situation. Section 3 describes the materials and methods used in our study. Section 4 presents the results and offers a more extensive explanation of our findings. Finally, conclusions and future areas for study are provided in Section 5.

[1] https://en.wikipedia.org/wiki/Category:Main_topic_classifications

2. Background

The scope of this paper is twofold: firstly, automatic domain-classification of LOV vocabularies according to Wikipedia categories, and secondly, the use of Deep Learning techniques to implement an effective document classifier. We provide an extensive bibliography on both domain-dependent document categorisation and Deep Learning classifiers. We also present a summary of relevant articles.

Document categorisation, defined in [9] as content-based assignment of one or more predefined categories to a document, has been extensively covered in several papers. The preprocessing tasks are studied in [10], which measures their impact in document categorisation. As well as the preprocessing stage, another important part of document categorisation is the use of a proper dataset for training. For this purpose, normalisation, stop word removal and stemming are combined to analyse which performs best. In [11], a dataset comprising 100 audiobook reviews is classified, evaluating three aspects: story, performance, and overall quality. In [12], two document classification methods called SemCla (Semantic Classifier) and SemCom (Committee with Semantic Categorizer) are proposed. These classifiers are based on semantic similarity and use an algorithm called SemCat (Semantic Categorisation) presented in a previous work by the same researchers. A text classification of a student's dataset is carried out comparing the accuracy of Naive Bayes classifier and K-Nearest Neighbor (KNN) classifier [13]. Finally, two survey papers in document/text classification are found in [14] and [15].

Document categorisation is also used in the field of Semantic Web, which means working with ontologies, vocabularies or Linked Data datasets. For example, [16] presents a classification of LOD datasets based on the different categories presented in the LOD cloud diagram. The most similar to our paper are [17] and [18]. The first presents a framework called OntClassifire, which makes use of a domain ontology to define the categories and benefits of ontology-matching techniques to classify 34 instances. The second describes a portal called OntoKhoj that searches, aggregates, ranks and classifies ontologies. It uses traditional algorithms for classification such as Naive Bayes, Term Frequency–Inverse Document Frequency (Tf-idf), Probabilistic Indexing (PRIND) and KNN, classifying 22 ontologies in five different domains. In most of the papers, ontology classification is used as: 'A way to compute a partial ordering

or hierarchy of named concepts in the ontology using the subsumption' [19]. In the present paper, the approach for classifying an ontology within a particular domain will be referred as ontology categorisation or classification.

The use of Wikipedia for categorisation can be found in [20]. This presents a demo where educational datasets from the Linked Data cloud are categorised into topics from DBpedia, the structured data version of Wikipedia [21]. Also, [22] makes use of Wikipedia to extend hierarchical classification with an unsupervised model called Folk-Topical Text categorisation (FTTC). Wikipedia is also used in [23], which presents a new text classification technique using an associate network. Associate networks allow users to analyse texts and find key concepts. In [24], Wikipedia is used for enriching the semantic information documents in Traditional Open Directory-Project (ODP), which is a text classification method. Finally, [25] presents a supervised text classification method in which the training sample is extended using Wikipedia concepts. This makes it easier to annotate the training data, which is therefore less time consuming.

The first Deep Learning models date from 1980, when Fukushima's Neocognitron was published [26]. Its first successful application with a high accuracy rate in a real use case took place in 1985 [27]. These techniques have been completely revolutionised in the last years, as reviewed in [28]. The landmark moment occurred during the 2012 ImageNet challenge[2], when a model's error rate was improved more than 10 per cent in image classification, [29]. These techniques have also obtained good results in several areas such as computer vision [30] or Natural Language Processing (NLP) [31], which is the field of this experiment.

Finally, some papers that benefit from Deep Learning techniques in document categorisation are summarised. A Convolutional Neural Network (CNN) is presented in [32]. The approach introduces the use of rationales for text classification. Another approach using Deep Learning for text classification is used in [33]. In this, an approach called Hierarchical Deep Learning for Text classification [3] (HDLTex) classifies documents, both complete or fragments, depending on the hierarchy level. Another method for text classification can be found in [34]. Here, three multi-task architectures of Recurrent Neural Networks (RNN) are used to classify four text benchmarks. Also, in [35], a model using CNN based on the attention model is used for text classification in mathematics. Finally, in [36], CNN are used to classify DBpedia.

Several differences are found when comparing this paper with those previously listed in this section. In the context of document classification or categorisation, only a few papers classify ontologies by domain, which are rarely the main objective of the experiments. It appears that no articles currently exist that assess the use of Deep Learning techniques for categorising ontologies by domain.

3. Materials and methods

This paper classifies LOV vocabularies by domain, using the main categories of Wikipedia and Deep Learning models. The experiment is divided into two principle steps: first, vocabularies were preprocessed, so they can be fed into the classifier, and second, the model was built, trained, tested and used. When preprocessing the data, it was first necessary to obtain the classes and properties from each vocabulary from LOV. This task was accomplished using RDFLib[4], a Python library that works with RDF. In the second step, a Deep Learning model was built and used as an ontology classifier in a two-step process: training and validation. To train the Deep Learning model, a corpus of tagged documents was required. In this case study, the corpus was formed by abstracts extracted form DBpedia categorised using Wikipedia categories. Fig. 1 shows the workflow of the whole process.

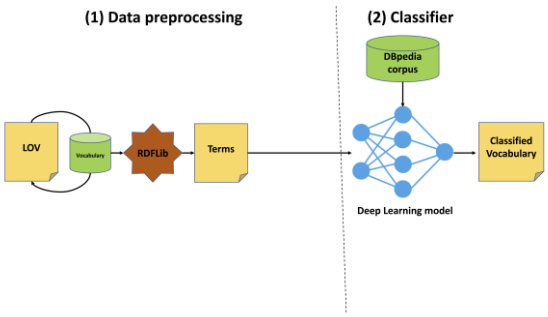

Fig. 1. Workflow followed to classify vocabularies.

[2] http://www.image-net.org/challenges/LSVRC/
[3] https://github.com/kk7nc/HDLTex
[4] https://github.com/RDFLib/rdflib

3.1. Preprocessing data in vocabularies

For that purpose, the dump provided by LOV[5] has been used. It includes the terms and characteristics of each vocabulary. One of the characteristics of the vocabularies is the usage of the tag 'keyword' which belongs to Data Catalog Vocabulary[6] (DCAT). By using this tag, vocabularies are assigned to a domain. As previously mentioned, LOV collaborators assign this tag, which raises the problem of introducing personal biases based on their background. For this reason, we chose to use Wikipedia categories.

First of all, the terms used in the classifier had to be extracted from each vocabulary. In this instance, the classes and properties of each vocabulary were selected.

The starting point was a downloaded dump from LOV that contained all the vocabularies with their terms: classes and properties. On 2nd November 2018, LOV contained 651 vocabularies saved as .n3[7] files, also known as Notation 3 files, a superset of RDF. In total, the dataset of vocabularies had a size of 48.3 megabytes. Then, the terms from each vocabulary were obtained using RDFLib.

In this step, there was a list of terms for each vocabulary: the classes and properties that have been obtained. Each list of terms comprised the input of our classifier, obtaining a category associated with the whole set. Next, each vocabulary had to be encoded for introduction into the model. For that purpose, word embeddings were used [37]. This is a means of representing text as a vector space. Here, a dictionary was created taking into account the total amount of words in a set of texts. The dictionary consisted of a list of words ordered by index according to the frequency with which they occur. For example, the word that appears most frequently in all texts occupies the first position, and so on. Then each text is codified by attributing the numerical position they have in the vocabulary to all the words of these texts. In this case based on [38], the study's dictionary contained 20,000 words and each text were codified in 400 words. This means that only the 20,000 most used words were used to build the dictionary. Also, when a text has less than 400 words, the rest of the vector was filled with zeros. Also, when a text had more than 400 words, only the first 400 were used to codify the text.

[5] http://lov.okfn.org/lov.n3.gz
[6] http://www.w3.org/TR/vocab-dcat/
[7] https://www.w3.org/TeamSubmission/n3/

3.2. Building the classifier

The next step was building the ontology classification model. As previously mentioned, Deep Learning techniques will be used. These kinds of models consist of three stages: training, validation and prediction (in this case, the categorisation of the vocabularies). For the training and validation dataset, we gathered a corpus of categorised documents. Finally, a model was built and used to predict the category of LOV vocabularies.

3.2.1. Gathering the corpus

A corpus of classified documents was needed in order to train and validate the model. The documents were tagged with Wikipedia's main topic categories. These classifications have 12 main categories, plus subcategories, which come to a total of 22. The 12 main categories and their Wikipedia descriptions are the following:

- Reference: this is for reference works considered a compendium of information, usually of a specific type, that are compiled in a book for ease of reference.
- Culture: refers to human activity; different definitions of culture reflect different theories for understanding, or criteria for evaluating, human activity.
- Geography: study of the earth, its features, inhabitants and phenomena.
- Health: the functional or metabolic efficiency of a living organism.
- History: the interpretation of past events, societies and civilisations.
- Mathematics: the study of topics such as quantity (numbers), structure, space, and change.
- Nature: a rational approach to the study of the universe, understood as obeying rules or laws of natural origin.
- People: refers to a general group, such as all humans, an ethnic group or a nation.
- Philosophy: encompasses all of knowledge and all that can be known, including the means by which such knowledge can be acquired.
- Religion: the adherence to codified beliefs and rituals that generally involve a faith in something of a spiritual nature, and the study of inherited ancestral traditions, knowledge and wisdom related to understanding human life.
- Society: refers to a large group of people sharing their own culture and institutions.

– Technology: an expanded concept that deals with a species' usage and knowledge of tools and crafts, and how it affects a species' ability to control and adapt to its environment.

Based on the previous domains, a Python scraper was built to create the tagged corpus. The scraper obtains the text, which consists of abstracts of articles from DBpedia. These were then categorised according to the main topic categories. The only category that was not considered is Reference, because it is not directly related with a particular field. To extract information from DBpedia, SPARQL[8] queries – the query language of the Semantic Web – were required. Each query obtained the abstract for each article by using the subject that corresponds to one of the categories. Then the broader categories were queried to obtain categorised abstracts. The hierarchy of categories had to be taken into account: an article in a broader category was also part of a main one. This process was repeated until a sufficient number of documents were scraped. The following two pieces of code are the queries used to obtain text related with 'Culture':

```
(1)
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX dbpedia-owl:<http://dbpedia.org/ontology/>
PREFIX dbc: <http://dbpedia.org/resource/Category>

SELECT DISTINCT ?resource, ?abstract WHERE {
   ?resource dcterms:subject dbc:Culture .
   ?resource dbpedia-owl:abstract ?abstract .
   filter langMatches(lang(?abstract),"en")
}
LIMIT 10000 OFFSET 0

(2)
PREFIX
skos:<http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT ?broader_cat WHERE {
   ?broader_cat skos:broader dbc:Culture .
}
LIMIT 10000 OFFSET 0
```

For each category, we attempted to download at least 50,000 documents. However, there were fewer documents available in the Culture category: specifically, 36,960. Looking at the documents, it was apparent that some contained very few words. This could lead to a training dataset with insufficient information.

A corpus must fulfil two criteria: it must be balanced, and it must have sufficient representativeness. A corpus is balanced when it contains a wide range of text genres that exist in the target language (categories in that case) [39]. According to [40], representativeness of a corpus is determined by 'the extent to which a sample includes the full range of variability in a population'. In order to balance the corpus, we obtained statistics in order to establish a minimum length of words per document. Based on Table 1, which lists the mean number of words per category, a minimum of 120 words was established. The table also provides information on the number of documents with 120 or more words, and the total number of documents and words in the corpus. Finally, the number of documents per category should be the same. As the category with fewest documents is 'People', this amount was established as 14,000.

Table 1

Corpus statistics.

Metric	Value
Nature documents	18,805
Mean words in Nature	120.10
Mathematics documents	18,816
Mean words in Mathematics	119.74
Society documents	21,182
Mean words in Society	137.51
Religion documents	20,823
Mean words in Religion	138.31
People documents	14,074
Mean words in People	105.07
Technology documents	21,378
Mean words in Technology	136.11
Philosophy documents	20,188
Mean words in Philosophy	132.72
Geography documents	19.130
Mean words in Geography	127.06
Health documents	19,810
Mean words in Health	130.50
Culture documents	15,845
Mean words in Culture	138.51
History documents	20,886
Mean words in History	140.36
Total amount of words	1,563,388,971,685
Number of documents	380,096

Once the corpus was compiled, we ensured that each category was sufficiently representative. An algorithm called Tf-idf [41] was used to accomplish this, by calculating which words were relevant in a document or a small group of

[8] https://www.w3.org/TR/rdf-sparql-query/

Table 2

Results after applying Tf-idf

Category	Word example	Weight	Description
Nature	Opossum	0.1391	A marsupial
Mathematics	Combinatorics	0.2101	An area of mathematics
Society	Baloch	0.1184	People who live in Balochistan
Religion	Psilocybe	0.1571	Gilled mushroom used for religious communion
People	Landulf	0.1171	A masculine given name
Technology	Talkboy	0.1237	Portable cassette player and recorder
Philosophy	Lycan	0.1156	Refers to William Lycan, American philosopher
Geography	Rujm	0.1544	An ancient megalithic monument
Health	Recessive	0.1102	A type of gene
Culture	Vestment	0.1632	Liturgical garments
History	Nengō	0.2717	A Japanese term for a calendar period of time

documents over an entire corpus. Mathematically it can be depicted as Eq. (1):

$$a_{ij} = tf_{ij} * \log(\frac{N}{n_j}) \quad (1)$$

Where tf_{ij} is the frequency of term j in document i, N is the total number of documents in the corpus and n_i the number of documents containing term j.

This gave a result between 0 and 1 measuring the importance of the word with respect to the rest of the corpus. Before applying Tf-idf, stop words and words that include numeric symbols and letters that do not belong to the Latin alphabet were removed. The algorithm was applied to the corpus and the five most relevant words for each category were obtained: the results can be found in Table 2. The first column displays the category in the corpus; the second is one of the words in the top five; the third, its weight; and the last column has a description of the word. As can be seen, the words that are relevant to the category are a good representation. So, it can be concluded that the corpus was representative for each category.

3.2.2. Building the model

Once the training and validation set was compiled, it was time to build the model. Random Multimodel Deep Learning (RMDL) was used, [42] and [43]. This is a hybrid model that combines Deep Neural Network (DNN), CNN and RNN. CNN has been widely used for text classification. Also, RNN was recommended in numerous papers, which makes more sense when the order of the words in the text is important [44]. The model was built using Keras[9], a high-level neural networks API that allows other neural network libraries such as TensorFlow[10] to be used on top, and this is the one chosen for this paper. In this instance, the first layer of the model is embedding: it receives as input a word embedding representation of the training data and has been pretrained with a word vector called GloVe [45]. Then, a hybrid model mixing three different Deep Learning models was built. It comprised a DNN model, a CNN and an RNN. Once the architecture was defined, the model was trained with 80 per cent of the data in 200 epochs with batch sizes of 16. The rest of the corpus was used for validation. After the training stage, the model showed a loss of 0.07 and an accuracy of 0.93. The metrics at validation time were 0.03 and 0.97.

3.2.3. Evaluation of the model

In order to evaluate the model, it was compared with a set of baseline models such as Support Vector Machine (SVM), Naive Bayes and Stochastic Gradient Descent (SGD). [46] describes SVM, a pattern classifier based on statistical techniques. This classifier finds a separating hyperplane that divides a dataset distributed in an n-dimensional space into classes. An SVM model was developed with Scikit-learn, a Python library for data mining and data analysis [47]. After using the same dataset with the SVM model, it displayed an accuracy of 59.2 per cent, which is [48] less accuracy than the neural model proposed in the paper. The dataset was also tested using Naive Bayes. Specifically, the Multinomial Naïve Bayes (MNB) was used for text categorisation. This classifier is based on the idea that a document belongs to a class depending on the probability that

[9] https://keras.io/

[10] https://github.com/tensorflow/tensorflow

Table 3

Comparison between models

Model	Accuracy (%)	Model
Support Vector Machine	59.2 %	Support Vector Machine
Multinomial Naive Bayes	62.4 %	Multinomial Naive Bayes
Stochastic Gradient Descent	60 %	Stochastic Gradient Descent
Random Multimodal Deep Learning	93.57 %	Random Multimodal Deep Learning

several words occur in a document from a category. Again, Scikit-learn was used to implement this model giving an accuracy of 62.4 per cent. Finally, the model has been evaluated against SGD, which is an iterative method that uses random examples of a training set to optimize a differentiable objective method [49]. The model was implemented again using Scikit-learn, obtaining an accuracy of 60 per cent. Table 3 shows a summary of the accuracy obtained from the different models. It can be seen that the Deep Learning model is by far the most accurate.

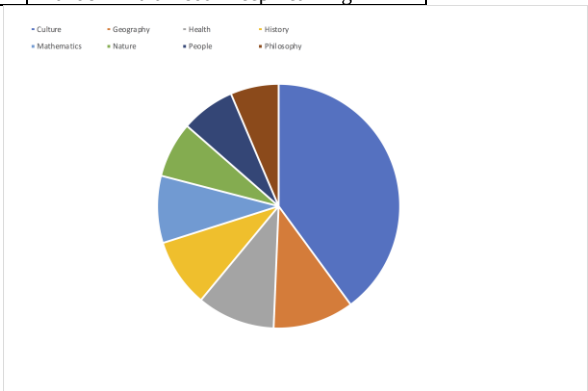

Fig. 2. Distribution of the vocabularies across different categories.

4. Results

In this section, the results of the experiment will be considered in depth. This section will be divided into subsections: one stage with the information obtained during the classification stage and a second one with the limitations of the study.

4.1. Classification of vocabularies

As previously stated, the main aim of this paper is to make an automatic classification of the vocabularies compiled in LOV. After running the model, the number of vocabularies that belong to the 11 main Wikipedia categories were obtained. Table 4 shows the results after the classification: the first column shows the category; the second, the number of vocabularies that belong to that category, and third, the percentage of vocabularies that belong to it. The output of the model was considered to belong to a single category, which, in this instance, was the one with the highest value in the output vector. The vocabulary was not classified in any category only when the values of the output were very widely distributed across different categories. In line with these criteria, 582 vocabularies have been categorised. Figure 2 demonstrates the distribution in a pie chart.

4.2. Limitations

Some limitations became apparent during the research for this paper. Firstly, some vocabularies could not be processed, either because they contained no terms or because the information could not be retrieved using RDFLib. In total, 72 vocabularies were discarded. In some instances, there were no terms to be retrieved. Others were not considered to belong to any of the categories, as the output of the model was widely distributed in percentage between the different categories. Secondly, only one-word terms were taken into account when obtaining the terms. For example, terms like 'accountServiceHomepage' were split and counted as three different words. This entirely changes the way information is preprocessed and how the classification is made. The third and final limitation is that only 11 very general domains were used for the classification. This means that it is impossible to go into depth into the classification, and only a general use of the vocabularies is provided to the user.

Table 4

Classification of vocabularies between categories

Category	Number of vocabularies
Culture	211
Geography	57
Health	55
History	48
Mathematics	47

Nature	39
People	38
Philosophy	34

5. Conclusions and future work

Two main issues have been addressed in this paper: first, a corpus with structured data from DBpedia has been obtained, and secondly, an automatic classifier was built and used. The corpus was obtained automatically by scraping abstracts from DBpedia using SPARQL queries. It was tagged according to Wikipedia's main categories. Finally, the main focus of the experiment, the Deep Learning model, has classified the vocabularies automatically.

Future works may include the use of n-grams in the preprocessing stage. This would ensure more accurate classification, and take into account, for example, words like 'accountServiceHomepage', which was mentioned in the Limitations section above.

The corpus could be extended by using the subcategories of Wikipedia's main categories. A corpus with a two-level hierarchy could be created, with one level comprising the eleven main Wikipedia categories, and a second level with its narrower subcategories. This would be useful for users who want to generalise with a vocabulary (those categorised in the first level) or to specialise (those classified in the second level).

Since the vocabularies used are ontologies, the work could benefit from their hierarchies. This means that the broader and narrower terms of each term could be used as the context for that term. This information could be applied when using word-embedding when codifying the vocabularies. In particular, a modification of Word2vec could be applied.

The application of RNN would benefit from context usage of each term. Also, it would be interesting to study terms from different vocabularies that have been categorised in different domains. When obtaining a classification with two levels, it would be interesting to make a comparison with the classification made by LOV, providing some mappings.

Finally, these kind of classifications are very useful for data retrieval strategies. For example, if a user needs to retrieve information about music and introduces the term 'bass' in a query, the search must be done in a dataset using vocabularies within Culture and not Nature, as 'bass' is the word for a fish as well as an instrument.

References

[1] Latif, A., Saeed, A. U., Hoefler, P., Stocker, A. & Wagner, C. (2009). The Linked Data Value Chain: A Lightweight Model for Business Engineers. 5th International Conference on Semantic Systems (p./pp. 568--575).

[2] Bizer, C., Heath, T., Idehen, K. & Berners-Lee, T. (2008). Linked data on the web. Proc. of the 17th Int. Conf. on World Wide Web (p./pp. 1265-1266): ACM. ISBN: 978-1-60558-085-2

[3] Lopes, N., Zimmermann, A., Hogan, A., Lukácsy, G., Polleres, A., Straccia, U. & Decker, S. (2010). RDF needs annotations. W3C Workshop on RDF Next Steps, Stanford, Palo Alto, CA, USA.

[4] Assaf, A., Senart, A. & Troncy, R. (2015). What's up LOD Cloud? Observing the State of Linked Open Data Cloud Metadata. In A. Rula, A. Zaveri, M. Knuth & D. Kontokostas (eds.), LDQ@ESWC, CEUR-WS.org.

[5] Bizer, C., Heath, T. & Berners-Lee, T. (2009). Linked data - the story so far. *Int. J. Semantic Web Inf. Syst*., 5, 1–22. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) *CONFERENCE 2016*, *LNCS*, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).

[6] Feitosa, D., Dermeval, D., Ávila, T., Bittencourt, I. I., Lóscio, B. F. & Isotani, S. (2018). A systematic review on the use of best practices for publishing linked data. Online Information Review, 42, 107-123.

[7] Vandenbussche, P.-Y., Atemezing, G., Poveda-Villalón, M. & Vatant, B. (2017). Linked Open Vocabularies (LOV): A gateway to reusable semantic vocabularies on the Web. Semantic Web, 8, 437-452.

[8] Bengio, Y., Courville, A.C., Goodfellow, I.J., & Hinton, G.E. (2015). *Deep Learning. Nature*, 521 7553, 436–44.

[9] Goller, C., Löning, J., Will, T. & Wolff, W. (2000). Automatic document classification: A thorough evaluation of various methods. 7. *Internationales Symposium für Informationswissenschaft.*

[10] Ayedh, A., Alwesabi, K., Rajeh, H., & Tan, G. (2016). The Effect of Preprocessing on Arabic Document Categorisation. *Algorithms*, 9, 27.

[11] Pappas, N. & Popescu-Belis, A. (2016). Human versus Machine Attention in Document Classification: A Dataset with Crowdsourced Annotations. In L.-W. Ku, J. Y. Jen Hsu & C.-T. Li (eds.), *SocialNLP@EMNLP* (pp. 94–100), Association for Computational Linguistics. ISBN: 978-1-945626-32-6

[12] Borkowski, P., Ciesielski, K., & Klopotek, M.A. (2017). Semantic classifier approach to document classification. *CoRR*, abs/1701.04292.

[13] Rajeswari R.P., Juliet, K. & Aradhana, Dr. Text Classification for Student Data Set using Naive Bayes Classifier and KNN Classifier. *International Journal of Computer Trends and Technology (IJCTT)* V43(1):8-12, January 2017. ISSN:2231-2803. www.ijcttjournal.org.

[14] Dhumale, I., Gupta, A., Gogawale, P., & Ranjan, N. (2015). Full Length Review Article *A Survey on Text Analytics and Classification Techniques for Text Documents*.

[15] Nalini, D. K. (2014). Survey on Text Classification. *International Journal of Innovative Research in Advanced Engineering*, 1, 412–417.

[16] Meusel, R., Spahiu, B., Bizer, C. & Paulheim, H. (2015). Towards Automatic Topical Classification of LOD Datasets. In C. Bizer, S. Auer, T. Berners-Lee & T. Heath (eds.), *LDOW@WWW*, CEUR-WS.org.

[17] Fahad, M., Moalla, N., Bouras, A., Qadir, M. A. & Farukh, M. (2011). Towards Classification of Web Ontologies for the Emerging Semantic Web. *J. UCS*, 17, 1021–1042.

[18] Patel, C., Supekar, K., Lee, Y. & Park, E. (2003). OntoKhoj A Semantic Web Portal for Ontology Searching, Ranking, and Classification. *Proc. 5th ACM Int. Workshop on Web Information and Data Management* (pp.58–61), New Orleans, Louisiana, USA.

[19] Kim, J.-M., Kwon, S.-H. & Park, Y.-T. (2009). Enhanced Search Method for Ontology Classification. *Computing and Informatics*, 28, 795–809.

[20] Taibi, D. & Dietze, S. (2016). Educational Linked Data on the Web – Exploring and Analysing the Scope and Coverage. In M. d'Aquin & D. Mroumtsev (eds.), Springer.

[21] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, Sö. & Bizer, C. (2014). DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*.

[22] Guo, N., He, Y., Yan, C., Liu, L., & Wang, C. (2016). Multi-Level Topical Text Categorisation with Wikipedia. *2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)*, 343–352.

[23] Bloom, N., Theune, M., & Jong, F.D. (2013). Using Wikipedia with associative networks for document classification. *ESANN*.

[24] Shin, H., Lee, G., Ryu, W., & Lee, S. (2017). Utilizing Wikipedia knowledge in open directory project-based text classification. *SAC*.

[25] Wenhao Zhu, Yiting Liu, Guannan Hu, Jianyue Ni, Zhiguo Lu. (2018). A Sample Extension Method Based on Wikipedia and Its Application in Text Classification. *Wireless Personal Communications Journal*.

[26] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36, 193-202.

[27] LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E. & Jackel, L.D. (1985). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1, 541–551.

[28] Bengio, Y. (2009). Learning Deep Architectures for AI. Foundations and Trends in Machine Learning, 2, 1–127.

[29] Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems (pp.1097–1105).

[30] Voulodimos, A., Doulamis, N., Doulamis, A. & Protopapadakis, E. Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, vol. 018, Article ID 7068349, 13 pages, 2018. doi:10.1155/2018/7068349

[31] Young, T., Hazarika, D., Poria, S. & Cambria, E. (2017). Recent Trends in Deep Learning Based Natural Language Processing. *CoRR*, abs/1708.02709.

[32] Zhang, X. & LeCun, Y. (2015). Text Understanding from Scratch (cite arxiv:1502.01710)

[33] Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S. & Barnes, L. E. (2017). HDLTex: Hierarchical Deep Learning for Text Classification. CoRR, abs/1709.08267.

[34] Liu, P., Qiu, X. & Huang, X. (2016). Recurrent Neural Network for Text Classification with Multi-Task Learning. CoRR, abs/1605.05101.

[35] Du, J., Gui, L., Xu, R. & He, Y. (2017). A Convolutional Attention Model for Text Classification. In X. Huang, J. Jiang, D. Zhao, Y. Feng & Y. Hong (eds.), NLPCC (p./pp. 183-195), Springer. ISBN: 978-3-319-73618-1

[36] Parundekar, R. (2018). Classification of Things in DBpedia using Deep Neural Networks. CoRR, abs/1802.02528.

[37] Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space (cite arxiv:1301.3781)

[38] Lenc, L. & Král, P. (2017). Deep Neural Networks for Czech Multi-Label Document Classification. CoRR, abs/1701.03849.

[39] Maekawa, K. (2008). Balanced Corpus of Contemporary Written Japanese. *ALR@IJCNLP*: Asian Federation of Natural Language Processing.

[40] Biber, D. (1993). Using Register-Diversified Corpora for General Language Studies. *Computational Linguistics*, 19, 219–241.

[41] Salton, G. & Buckley, C. (1988). Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24, 513–523.

[42] Kowsari, K., Heidarysafa, M., Brown, D.E., Meimandi, K.J., & Barnes, L.E. (2018). RMDL: Random Multimodel Deep Learning for Classification. *CoRR*, abs/1805.01890.

[43] Heidarysafa, M., Kowsari, K., Brown, D.E., Meimandi, K.J., & Barnes, L.E. (2018). An Improvement of Data Classification Using Random Multimodel Deep Learning (RMDL). *CoRR*, abs/1808.08121.

[44] Lai, S., Xu, L., Liu, K. & Zhao, J. (2015). Recurrent Convolutional Neural Networks for Text Classification. In B. Bonet & S. Koenig (eds.), *AAAI* (pp.2267–73).

[45] Pennington, J., Socher, R. & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. *EMNLP* (pp.1532–43).

[46] Boser, B. E., Guyon, I. & Vapnik, V. (1992). A Training Algorithm for Optimal Margin Classifiers. *Computational Learning Theory* (pp.144–152).

[47] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–30.

[48] Frank, E. & Bouckaert, R. R. (2006). Naive Bayes for Text Classification with Unbalanced Classes. In J. Fürnkranz, T. Scheffer & M. Spiliopoulou (eds.), *PKDD* (pp.503510): Springer. ISBN: 3-540-45374-1

[49] Robbins, H. & Monro, S. (1951). A Stochastic Approximation Method. *Annals of Mathematical Statistics*, 22, 400–4