# Subject Classification of Learning Resources Using Word Embeddings and Semantic Thesauri

Dimitrios A. Koutsomitropoulos
*Computer Engineering and Informatics Dpt.*
*University of Patras*
Patras, Greece
koutsomi@ceid.upatras.gr

Andreas D. Andriopoulos
*Computer Engineering and Informatics Dpt.*
*University of Patras*
Patras, Greece
a.andriopoulos@upatras.gr

Spiridon D. Likothanassis
*Computer Engineering and Informatics Dpt.*
*University of Patras*
Patras, Greece
likothan@ceid.upatras.gr

***Abstract* — Open Educational Resources (OERs) are often scattered among various sources and may follow different metadata schemata. In addition, they may not include exhaustive annotations; even worse, their subject characterization, if any, may be represented by arbitrary, ad-hoc keywords instead of standard, controlled vocabularies, a fact that stretches up the search space and hampers interoperability. To address this issue, in this paper we propose a twofold method based on two seemingly disjoint technology stacks: machine learning and the semantic web. First, OERs harvested from various repositories are assigned subject terms from a formal, standard thesaurus for a domain of interest, by discovering the semantic matches of the harvesting keyword within the thesaurus ontology. Then, we use word embeddings to represent an item's metadata and compute its similarity with the thesaurus keywords. These word embeddings are learned by a doc2vec model that has been trained with already annotated corpora from the biomedical domain. By combining both worlds, we show that it is possible to produce a reasonable set of thematic suggestions which exceed a certain similarity threshold.**

***Keywords—learning objects, OERs, classification, word embeddings, thesauri, ontologies, doc2vec, federated search***

## I. Introduction

Open Educational Resources (OERs) are becoming turnkey learning objects (LOs) during the last few years. In settings such as institutional libraries, repositories and information services the ability to combine and reuse such content is of major importance [1]. Despite research and methodology advancements in the field of metadata organization, it is often the case for OERs to be poorly or vaguely annotated. Moreover, the massive availability of OERs is rapidly adopting characteristics of Big Data [2]. Therefore, the task of discovering the most appropriate resources to glean together for e-learning purposes becomes cumbersome and error-prone.

On the one hand, educational resources are scattered among a variety of repositories and providers worldwide; on the other hand, they are frequently annotated using a proprietary metadata format, if any and, when it exists, their subject classification may follow un-authoritative keyword lists or ad-hoc vocabularies. To this end, the use of knowledge organization systems such as thesauri maintained by independent bodies and institutions is well-acknowledged, but not as frequently implemented. However, adequate content characterization using authority vocabularies requires elaborate, time-consuming, manual efforts, often involving field experts.

In earlier work we have shown that it is possible to harvest OERs from disparate providers in a federated manner and to fetch a least common subset of metadata elements based on a Learning Object Metadata (LOM) schema [3]. Selected resources can be kept in a local institutional infrastructure known as the Learning Object Ontology Repository (LOOR) for further reuse [4]. In this paper we argue that user keywords used to seed harvesting can be reused to thematically annotate learning resources. These keywords are matched against well-structured thematic thesauri, expressed in the Web Ontology Language (OWL) and then expanded based on their structure/semantic relations to boost recall of the search process. Discovered thesaurus terms can then be leveraged to thematically annotate selected OERs. To represent thesauri, we use the Simple Knowledge Organization System (SKOS) model, a Semantic Web standard that facilitates publication and use of thesauri as linked data [5].

To solidify this approach, we further verify and amend these semantic matches with additional thematic suggestions coming from a machine learning process that employs the doc2vec algorithm: thesauri terms tagging OERs are automatically learned using word embeddings of their title and abstract. By combining these two approaches we demonstrate that it is possible to produce a reasonable set of thematic suggestions which exceed a certain similarity threshold. The added benefit of the collaboration between the logical formalism of web ontologies and non-symbolic inference for subject classification of OERs lies in the heart of this contribution and, to our knowledge, has been seldom investigated before.

In the following, we first review related work in the field of Natural Language Processing (NLP) and machine learning methods for text classification. Next, in section III we present our methodology and architectural details for federated search and subject classification. Section IV describes the design of our experiments, datasets and thesauri used for evaluation as well as the baseline for comparison. Section V discusses results by testing our approach on a medical dataset using Medical Subject Headings (MeSH) [6]. Our conclusions and future work are summarized in the last section VI.

978-1-7281-1862-8/19/$31.00 ©2019 IEEE

## II. Background and Related Work

Artificial neural networks constitute a powerful tool for text analysis and classification. In particular, recent methods based on Convolutional Neural Networks (CCN) [7][8], Recurrent Neural Networks (RNN) [9] and Long Short-Term Memory (LSTM) [10] make use of deep learning to predict and classify information using natural language data as input.

To represent text data, the word embeddings technique can be used: words and phrases are mapped into vectors of real numbers. This encoding may be simple, e.g. every word from a sentence is represented by a different number. A more intelligent approach, known as word2vec, was proposed by Mikolov [11] with two models of architecture. In this method, vectors come as a result of training a shallow neural network, and it is possible to examine syntactic and semantic similarities by vector comparison. In a similar manner, doc2vec computes vectors for entire documents or paragraphs rather than mere keywords [12]. Related studies use these particular models independently as well as in combination with others in various text classification tasks [13][14][15][16], sentiment analysis [17], question and answer retrieval from text corpora [18], even use of Principal Component Analysis (PCA) [19].

A demanding task is to assign a subject to a collection of OERs. Data stored in public repositories are fully open access to everyone, however, they do not always have a valid or formal subject annotation to make this access easier for the end user. So far, there have been relevant studies which, with the use of word embeddings and the PageRank algorithm [20][21][22], present a framework for automatic extraction and ranking of keywords. Also, another study extends word embedding models and employs the simple k-nearest neighbor search to predict tags for unseen documents [23].

To the best of our knowledge, the combination of word embedding techniques, in particular doc2vec, with ontology-based semantic matching and expansion for subject classification of OERs has not been proposed before. Moreover, the overwhelming majority of studies attempt and eventually classify the data uniquely into a finite and quite small number of categories, also possessing a large number of samples per category. In the present study, however, the subject assignment is done with the help of a thesaurus, which contains several thousands of labels. Furthermore, each sample is categorized by assigning more than one label to it.

## III. Design and Methodology

### A. Federated Search

To address metadata incompatibilities between OER repositories the creation and maintenance of a semantics-aware Learning Object Ontology Repository (LOOR) has been proposed in [4]. Such a repository can tap into ontologies and thesauri and allow LO metadata instances be assigned machine-understandable semantic annotations. Building on this premise, LOs can be ingested, and different schemata be aligned within the LOOR into a common LO ontology. An outline of this ontology is shown in Fig. 1, combining terminology from the LOM standard and Dublin Core.

First, a federated query is initiated towards the various repositories. Next, the metadata of items returned as responses to the query are harvested and aligned to a unified LO Ontology Schema, using a common set of elements and mapping rules. The subject of these items is then automatically populated taking as basis the initial query keywords in accordance with thematic thesauri for specific knowledge domains. At this point, a curator or instructor may review the LO and decide to incorporate it into the LO ontology, thus letting it available for others to reuse.

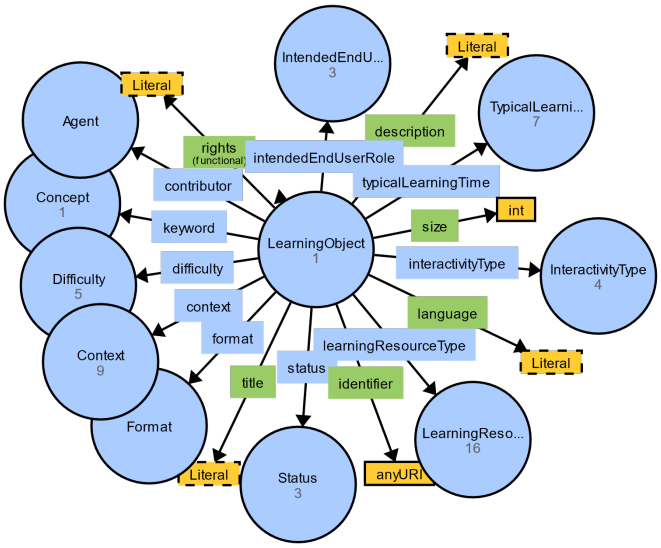


Fig. 1. Visualization of the LO Ontology Schema classes and properties.

Currently, data sources include MERLOT II, a large archive of OERs [24], Europe PubMed Central, a major repository of biomedical literature [25], ARIADNE finder, a European infrastructure for accessing and sharing learning resources [26] and openarchives.gr, the entry point for Greek scholarly content.

### B. Match SKOS Thesauri

The purpose of query expansion when harvesting OERs has been investigated before by the authors [3]. In essence, keywords that initiate harvesting are matched against expert terminological knowledge expressed in the form of term thesauri following the SKOS model in OWL format. Each keyword can thus be expanded into several narrower keywords that refine the former by performing reasoning about the semantic relationships of the matching terms in the thesaurus hierarchy. For example, the SKOS relations *skos:broader / skos:narrower* are used to connect a concept with its refinements. Further, a thesaurus term can be comprised of multiple lexical representations, including alternative labels and translations in different languages, as represented by the properties *skos:preflabel* and *skos:altlabel*.

In this paper, we maintain and reuse the information caused by the exploration of the thesauri term hierarchy in order to thematically annotate an item, when it is selected for addition into the LOOR. As a result, original search keywords are being repurposed to provide subject annotations for selected LOs. Merely suppling arbitrary keywords as subjects would not make much sense; rather, these keywords are first matched and refined against formal thematic thesauri and the matches are injected as semantic subject annotations into the selected OERs, using the *lom:keyword* property of the LO Ontology Schema.

As an example, consider the seed keyword "medicine". This is matched by a homonym term in the thesaurus and refined for example by the term *pathology.* Results matching the various labels of *pathology* will be automatically classified with the concept *pathology* as well as the concept *medicine*, since this is the topmost parent matching term for the initial keyword.

### C. Word Embeddings

In our work we employ the doc2vec model, namely the Distributed Memory architecture, which achieves the prediction task of the next word in a context to create a dictionary, trained with material from OER corpora. Our aim is to create a model that would learn as many terms as possible from a vocabulary that represents a formal domain of knowledge, i.e. a thesaurus. Then, given the title and the abstract of a learning resource, the model would be able to predict those thesaurus terms that most closely represent the subject of the resource. The title and abstract are the two annotations of the least common subset of elements stored in the LOOR with the richest semantic meaning about the resource. The other would be the resource itself (i.e. the full text). Keywords are also capable of conveying semantics but, as discussed, they are optional, frequently ad-hoc and are not guaranteed to be precise or to reference standard vocabularies. Moreover, the doc2vec algorithm operates on entire paragraphs or phrases instead of words. However, proper classification keywords (thesaurus subject annotations) do exist in our training set and they are used to tag paragraphs during training.

The title together with the abstract from each OER form a single *body of text*. Next, we convert this body into a list of lowercase tokens, while words with length shorter than two as well as special symbols are removed. For our dataset of short titles and abstracts, we have seen that retaining stop words generally improves similarity scores. Every title and abstract text is now a list of words. In our training dataset, every OER has been already annotated and classified by experts using terms from an appropriate thesaurus, e.g. MeSH for the biomedical field. Each such term in the thesaurus has a unique ID, so we select this ID instead of text to deal with multiple lexical representations of a term. As a result, each body text (title and abstract) is tagged by one or more term IDs, one for each term that occurs as its subject annotation in the dataset.

During training, the vector of each thesaurus term appearing in the dataset embeds information related to the entire abstract and title, thus each body text will be assigned a tag from one or more of these terms. However, it is unlikely that every possible term of the thesaurus would occur in the dataset, so there is a chance that several terms might be missing from the learned dictionary. To compensate for this loss, information from the thesaurus is also integrated into the training phase. In particular, the text of the term description is also used for training. In this way, almost complete coverage of the dictionary's words is achieved.

The trained model can answer queries about the similarity of whole texts and words. In addition, given a text, it detects a predetermined number of related words, calculates their similarity and sorts them in descending order.

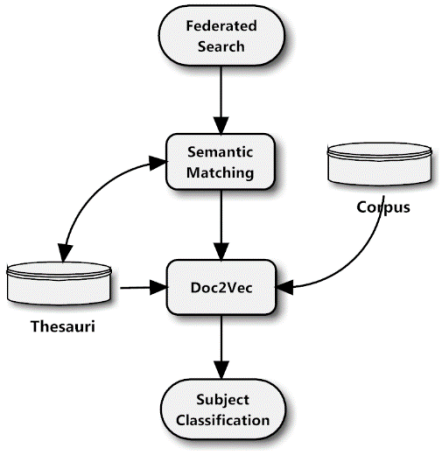


Fig. 2. Overall system process flow.

### D. Integrating semantic annotations and word embeddings

Both approaches described previously work collaboratively to provide subject classification suggestions, following the process flow shown in Fig. 2. First, metadata about OERs are harvested from the remote repositories into the LOOR and mapped to the unifying ontology schema. Then, semantic subject annotations are injected into these metadata, based on the seed keywords and term matching and expansion within the thesaurus ontology (*Semantic Matching*).

Next, the title and abstract of each OER metadata are fed into the trained doc2vec model and are corresponded to a single vector (*Doc2Vec*). The subject terms which have already been injected are searched for in the model dictionary, using the term IDs. Terms not occurring in the dictionary are ignored. Based on the model's output, each term is assigned a similarity score, as a means to assess the quality of the term suggestion.

Furthermore, the trained model also seeks similarities between the OER and other thesaurus terms contained in the dictionary and outputs the top 10 terms with the highest similarity score. These terms can be further considered by a curator for inclusion when adding the OER into the LOOR. Naturally, the similarity of a proposed term to an item's body text, no matter where does it come from (semantic matching or doc2vec itself), is a measure of the quality of this suggestion. Therefore, it might be useful to set a threshold above which suggestions are retained or discarded otherwise.

An example demonstrating the subject classification and scoring scheme discussed before is depicted in Fig. 3. A specific OER returned by federated search (https://doi.org/10.1007/s11657-019-0590-5) is annotated with two terms by the semantic matching process. Then, doc2vec computes their similarity scores and proposes another 10 subject terms along with their score.

## IV. Experimental Setup

### A. Evaluation procedure

To evaluate our methodology, we conduct three representative experiments. First, we test the trained doc2vec model against part of the training set [27]. This is reasonable,

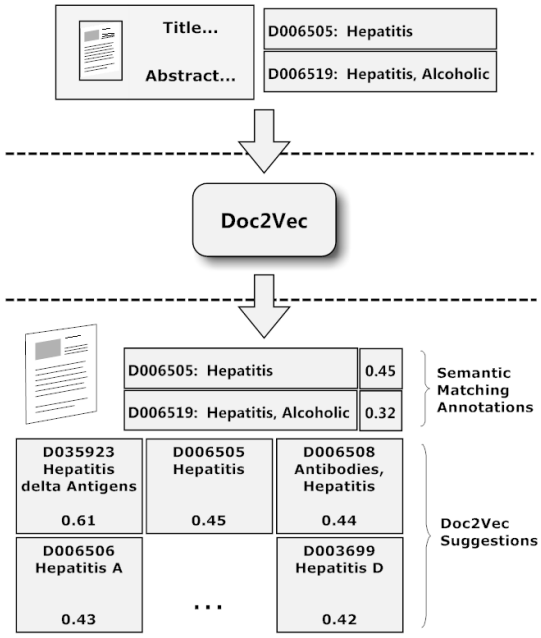


Fig. 3. A specific item gets subject annotations and their similarity scores are computed.

since doc2vec can only perform well with texts and words already contained in its dictionary.

Second, we test the model with another, unknown test set and see how it performs when it is supplied with arbitrary titles and abstracts. In contrast to word2vec, doc2vec is capable of inferring vector representations of body texts not presented before to the model. The results of this second experiment are typical of a federated search scenario with arbitrary keyword seeds and, therefore, represent our baseline or threshold above which term suggestions can be retained.

Third, we evaluate the quality of the semantic matching suggestions by computing their average similarity. In addition, we present the average similarity score of the best suggestion made by doc2vec itself, for the purposes of comparison.

The metric used for the model evaluation is the cosine similarity [11]. It is a widespread method in information retrieval and related studies. It accepts two vectors as arguments and returns the measure of their similarity in values included in the closed interval between zero and one. The higher the measure, the greater the similarity. Given two vectors $w_1$ and $w_2$ the formula that defines the relation is:

$$Similarity(w_1, w_2) = \cos(\theta) = \frac{\overrightarrow{w_1} \cdot \overrightarrow{w_2}}{|\overrightarrow{w_1}| \cdot |\overrightarrow{w_2}|}$$

The similarity score is computed for each sample of the test sets used in the three experiments and the average performance of the model in all samples is reported.

Doc2vec training has been performed using the following parameters: *train epochs* 10, *size vector* 100, *learning parameter* 0.025, and *min count* 10. The created model is saved so that it can be called directly when appropriate.

### B. Dataset

For the application of the doc2vec method, a dataset from the PubMed[1] repository with records of biomedical citations and abstracts was used. In December of every year, the core PubMed dataset integrates any updates that have occurred in the field. Each day, the National Library of Medicine produces update files that include new, revised and deleted citations. Each entry in the dataset contains information, such as the title and abstract of the article, the journal which the article was published in, and a list of subject headings that follow the MeSH thesaurus.

MeSH is a formal, specialized thematic thesaurus that gives uniformity and consistency to the indexing and cataloging of biomedical literature. MeSH has been already implemented using the SKOS vocabulary specification into OWL format [28]. It is a relatively large and dense thesaurus, comprising of 23,883 SKOS concepts (thesaurus terms).

The data is available in XML format [29]. The elements finally used for doc2vec training are *ArticleTitle*, *AbstractText* that represent the body text; and, from the *MeshHeadingList*, the *DescriptorName* with *MajorTopicYN* = *"Y"* or the *QualifierName* with *MajorTopicYN* = *"Y"* that represent the tags.

The file used for training contains 155,963 samples (bibliographic items). These samples contain a total of 420,165 MeSH terms, i.e. an item may be annotated with multiple terms, while the unique terms are 11,686, which cover 49% of the total vocabulary of 23,883 words. For the sake of completeness, 11,883 additional terms were selected from the thesaurus file. The selection criterion is for these terms to have descriptions, specifically the *scopeNote* field, roughly representing a brief definition of the term. In total we have covered 99% of the dictionary since we have 23,569 unique terms. However, for these additional terms, the model is trained using only a single body text which is the contents of the scopeNote field; therefore, such terms may not be adequately learned yet.

## V. Results and Discussion

Initially, we select the doc2vec model which was derived from the word embedding process and to which unsupervised training was applied. Evaluation is carried out by checking the similarity of text sentences among a dataset of 15,383 items, a subset of the training set. With a repetitive procedure we calculate the average similarity score. Specifically, we create the vector of the title and the abstract text by supplying this body text as an argument to the model, while we draw the vector which already exists in the model's dictionary using the term ID. These two elements normally have a high degree of correlation. Then, through the metric of the cosine similarity between two vectors, we calculate the similarity between the two. Because each item can be annotated with multiple terms, the repetitions executed reach the amount of 46,693. The final results are depicted in Fig. 4. The mean and the standard deviation of the results are 0.43 and 0.12 respectively.

An additional test is performed to check the reliability of the model on a different dataset, unknown to the model. This dataset is comprised of another 13,470 items that have not been used for

[1] https://www.nlm.nih.gov/databases/download/pubmed_medline.html

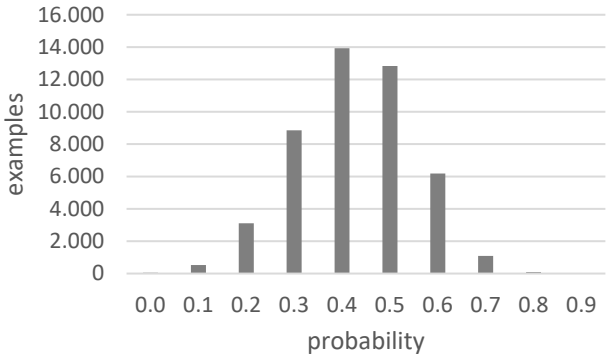


Fig. 4. Ranking of 46,693 examples by similarity

training. The terms that annotate these items are contained in the first 49% of the total vocabulary that has been learned from the PubMed training dataset. The total annotation count is 41,374.

The results are of interest as, even with unseen body texts, the model responds satisfactorily regarding their similarity, as depicted in Fig. 5. The mean and the standard deviation of the results are 0.36 and 0.12 respectively. We notice a slight drop in the average similarity score due to the dataset being unknown. However, we select this score as our threshold for term selection, considering the worst-case scenario where the model is oblivious to the body texts used as inputs.

Finally, another test is performed using a dataset of 1,405 items. These items have been specifically returned by the federated search procedure and their annotations are produced by Semantic Matching. The mean and the standard deviation of the average similarity scores for these annotations are 0.30 and 0.13 respectively (case 1). Out of 1,805 annotations, 596 (33%) pass the 0,36 threshold posed before for arbitrary texts and can be ultimately retained when selecting the item for addition into the LOOR.

Next, we let the model produce its own suggestions asking for the top 10 terms with the highest similarity score (case 2). The best suggestion made has a mean of 0.58 and a standard deviation of 0.07 (case 2). The final results are depicted in Fig. 6. Additionally, the top 10 suggestions have a mean of 0.54 and a standard deviation of 0.07. In summary, the results of the above experiments are shown in Table 1.

A relatively increased similarity is noticed in comparison to all previous experiments. This is justified by the fact that the

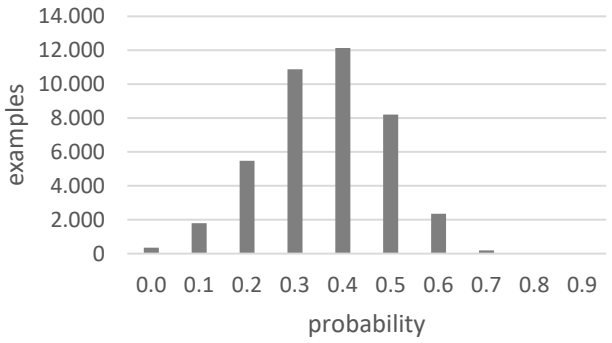


Fig. 5. Ranking of 41,374 examples by similarity.

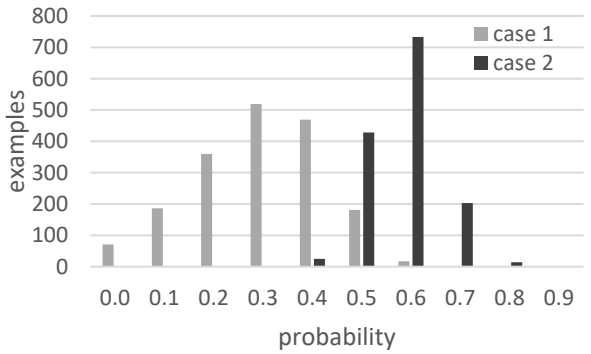


Fig. 6. Ranking of 1,805 examples by similarity.

model is no longer limited to checking specific annotations. Now it is given the freedom to choose the most similar vector from a rather large repository of 23,569 unique terms contained in its dictionary. Therefore, on account of the reliability offered by the doc2vec model as well as the data used for training, suggestions made by the model itself are determined by greater similarity.

TABLE I. EXPERIMENT RESULTS

| Metrics | Experiments with dataset | | | |
|---|---|---|---|---|
| | ***Training subset*** | ***Test set (unseen)*** | ***Semantic matching annotations*** | ***Doc2vec suggestions*** |
| Total annotations | 46,693 | 41,374 | 1,805 | 1,405 |
| Similarity Average | 0.43 | 0.36 | 0.30 | 0.58 |
| Similarity Stdev | 0.12 | 0.12 | 0.13 | 0.07 |

Given the difficulty of the problem of assigning a subject to an unstructured body of text and, in fact, from a large repository of unique terms, the results of the above experiments are considered satisfactory. Despite the scores being relatively not too high, exact values are not really critical. The model is able to assign similarity to annotations yield by semantic matching and make its own suggestions with even greater certainty. Moreover, the definition of the 0.36 threshold will assist in selecting or rejecting suggestions.

It is also evident that the performance of the model depends greatly on the training set. We might have circumvented the dictionary sparsity by including thesaurus terms with their descriptions directly from the thesaurus document, but this is still just one annotation for each term. Better results could not be achieved by simply addressing an even larger dataset; rather, the latter needs to be broad enough to cover as many terms as possible and to contain adequate samples for each term. This is by no means straightforward, since literature tends to concentrate on limited sets of concepts during the years.

## VI. CONCLUSIONS AND FUTURE WORK

Subject classification of OERs is a highly involving task, as it depends on several parameters ranging from availability of resources to metadata incompatibilities to intended OER use and synthesis. Reusing seed keywords can offer an alternative for missing or ad-hoc annotations; subject suggestions are authority controlled and refer to formal bodies of domain knowledge. In

addition, these suggestions can be assessed through a threshold posed by computing similarity between thesaurus terms and OER metadata. Not only can the construction of word embeddings for these two validate subject annotations but it can also make additional proposals for thematic classification.

As a next step, we intend to further evaluate our approach by involving human experts, that is, curators and instructors, during the OER selection and annotation process. Thus, we could verify exactly how many subject suggestions made by the system are actually satisfactory and if the currently selected threshold actually matches user needs. Additionally, a much larger dataset, even from different repositories, could be incorporated into the training process. The use of a distributed infrastructure could help with the increased needs for space and computational power that will be posed by such big data requirements.

Thesaurus terms in the dictionary are not independent of each other. There exist semantic relationships among them that for example may generalize a term into broader terms or specialize a term into narrower ones. Such prior knowledge might be useful to be incorporated into the model during training, including any arbitrary keywords that may already exist. In any case, the interplay between the logical formalisms of the Semantic Web and the learning capabilities of machine and deep learning form a line of research that is worth pursuing on.

## REFERENCES


[1] N. Piedra, J. A. Chicaiza, J. López, and E. Tovar. "An Architecture based on Linked Data technologies for the Integration and reuse of OER in MOOCs Context," Open Praxis 6 (2): 171-187, 2014.

[2] S. Eichhorn and G. W. Matkin. "Massive open online courses, big data, and education research," New Directions for Institutional Research, 2015 (167): 27-40. Wiley, 2016.

[3] D. A. Koutsomitropoulos, G. D. Solomou, and A. K. Kalou. "Federated Semantic Search Using Terminological Thesauri for Learning Object Discovery," International Journal of Enterprise Information Management 30 (5): 795-808. Emerald, 2017.

[4] D. A. Koutsomitropoulos and G. D. Solomou. "A Learning Object Ontology Repository to Support Annotation and Discovery of Educational Resources using Semantic Thesauri," IFLA Journal 44 (1): 4-24. SAGE, 2018.

[5] A. Miles and S. Bechhofer, eds. "SKOS Simple Knowledge Organization System Reference". W3C Recommendation, 2009. Available: http://www.w3.org/TR/skos-reference

[6] U.S. National Library of Medicine. "Medical Subject Headings", 2019. [Online]. Available: https://www.nlm.nih.gov/mesh/meshhome.html

[7] M. Hughes, I. Li, S. Kotoulas, and T. Suzumura, "Medical Text Classification Using Convolutional Neural Networks," in Informatics for Health: Connected Citizen-Led Wellness and Population Health, IOS Press, pp. 246-250, 2017.

[8] H. Xu, M. Dong, D. Zhu, A. Kotov, A. I. Carcone, and S. Naar-King, "Text Classification with Topic-based Word Embedding and Convolutional Neural Networks," in Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB 16), Seattle WA USA, pp. 88–97, October 2016.

[9] P. Liu, X.Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16), New York, USA, pp. 2873-2879, July 2016.

[10] P. Semberecki and H. Maciejewski, "Deep learning methods for subject ext classification of articles," in Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, pp. 357-360, September 2017.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient estimation of word representations in vector space." In ICLR Workshop, 2013.

[12] Q.V. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in 31st International Conference on Machine Learning, ICML, 2014.

[13] A. Mandelbaum and A. Shalev, "Word Embeddings and Their Use In Sentence Classification Tasks," in CoRR, vol. abs/1610.08229, 2016.

[14] R. A. Stein, P. A. Jaques, and J. F. Valiati, "An analysis of hierarchical text classification using word embeddings," Information Sciences, Volume 471, pp. 216-232, 2019.

[15] Q. Liu, H. Huang, Y. Gao, X. Wei, Y. Tian, and L. Liu, "Task-oriented Word Embedding for Text Classification." COLING, 2018.

[16] C. A. Turner, A. D. Jacobs, C. K. Marques, J. C. Oates, D. L. Kamen, P. E. Anderson, and J. S. Obeid. "Word2Vec inversion and traditional text classifiers for phenotyping lupus. BMC" in Medical Informatics and Decision Making, vol.17, pp. 126.-136, January 2017.

[17] R. Petrolito and F. dell'Orletta, "Word Embeddings in Sentiment Analysis," in Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), vol. 2253, Torino, Italy, 2018.

[18] R. Petrolito and F. D. Orletta, "Document retrieval and question answering in medical documents. A large-scale corpus challenge'', in Proceedings of the Biomedical NLP Workshop associated with RANLP, Varna, Bulgaria, pp. 1-7, September 2017.

[19] Z. Meilin, "Research on Text Classification Method Based on Multi-type Classifier Fusion," in 8th International Conference on Social Network, Communication and Education (SNCE 2018), Shenyang, China, vol. 83, pp. 798-805, May 2018.

[20] R. Wang, W. Liu, and C. McDonald, "Corpus-independent generic keyphrase extraction using word embedding vectors," in Software Engineering Research Conference, vol. 39, 2014.

[21] R. Wang, W. Liu, and C. McDonald, "Using word embeddings to enhance keyword identification for scientific publications," in Proceedings of the 26th Australasian Database Conference, ADC 2015, Melbourne, Australia. Springer, pp. 257–268, June 2015.

[22] D. Mahata, J. Kuriakose, R.R. Shah, R. Zimmermann, and J.R. Talburt, "Theme-Weighted Ranking of Keywords from Text Documents Using Phrase Embeddings," in IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, USA, pp. 184-189, April 2018.

[23] S. Chen, A. Soni, A. Pappu, and Y. Mehdad, "DocTag2Vec: An Embedding Based Multi-label Learning Approach for Document Tagging," in Proceedings of the 2nd Workshop on Representation Learning for NLP, Vancouver, Canada, pp. 111-120, August 2017.

[24] F. McMartin. "MERLOT: a model for user involvement in digital library design and implementation," Journal of Digital Information, 5 (3), 2006.

[25] Europe PMC Consortium. Europe PMC: A Full-Text Literature Database for the Life Sciences and Platform for Innovation. Nucleic Acids Research 43. Database issue (2015): D1042–D1048. PMC. Web. 11 Aug. 2017.

[26] S. Ternier, K. Verbert, G. Parra, B. Vandeputte, J. Klerkx, E. Duval, et al. "The ariadne infrastructure for managing and storing metadata," IEEE Internet Computing, 13(4), 2009.

[27] T. Schnabel, I. Labutov, D. M. Mimno, and T. Joachims, ''Evaluation methods for unsupervised word embeddings,'' in Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal, pp. 298-307, September 2015.

[28] M. Van Assem, V. Malaisé, A. Miles, and G. Schreiber. "A Method to Convert Thesauri to SKOS," In The Semantic Web: Research and Applications: 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings (Vol. 4011, p. 95). Springer, 2006.

[29] U.S. Department of Health & Human Services, "MEDLINE®PubMed® XML Element Descriptions and their Attributes," 2018. [Online]. Available: https://www.nlm.nih.gov/bsd/licensee/elements_descriptions.html#coistatement