# EMBEDD-ER : EMBEDDing Educational Resources Using Linked Open Data

Aymen A Bazouzi, Mickaël Foursov, Hoël Le Capitaine, Zoltan Miklos

# EMBEDD-ER : EMBEDDing Educational Resources Using Linked Open Data

Aymen A. Bazouzi[1][a], Mickaël Foursov[1][b], Hoël Le Capitaine[2][c] and Zoltan Miklos[1][d]
[1] *Univ Rennes CNRS IRISA, France*
[2] *Nantes Université, LS2N, UMR 6004, F-44000 Nantes, France*
{*aymen.bazouzi, mickael.foursov, zoltan.miklos*}*@irisa.fr, hoel.lecapitaine@univ-nantes.fr*

Keywords: Educational Resources, Embeddings, Linked Open Data.

Abstract: There are a lot of educational resources publicly available online. Recommender systems and information retrieval engines can help learners and educators navigate in these resources. However, the available educational resources differ in format, size, type, topics, etc. These differences complicate their use and manipulation which raised the need for having a common representation for educational resources and texts in general. Efforts have been made by the research community to create various techniques to homogeneously represent these resources. Although these representations have achieved incredible results in many tasks, they seem to be dependent on the writing style and not only on the content. Furthermore, they do not generate representations that reflect a semantic representation of the content. In this work, we present a new task-agnostic method (EMBEDD-ER) to generate representations for educational resources based on document annotation and Linked Open Data (LOD). It creates representations that are focused on the content, compact, and can be generalized to unseen resources without requiring extra training. The resulting representations encapsulate the information found in the resources and project similar resources closer to one another than to non-similar ones. Empirical tests have shown promising results both visually and in a subject classification task.

## 1 INTRODUCTION

The use of technology has become a necessity nowadays, whether it is for accomplishing tasks that were impossible to accomplish before or to facilitate and automate other tasks. Researchers have tried to make use of the recent technology breakthroughs and incorporate them in every field. The field of education is no exception, as it was made clear early on that this field has the potential to yield great results with meaningful impact on our lives. Technology can be used in many ways to assist educators and learners in enhancing their teaching and learning experiences. This gave birth to the term Technology-Enhanced Learning (TEL) (Kirkwood and Price, 2014).

TEL can be used in a lot of ways and in close collaboration with many fields. On the one hand, AI and data mining are among the most popular disciplines in the research community. On the other hand, the amount of educational content available is bigger than what humans can go through manually. Therefore, it quickly became apparent that education can benefit immensely from the use of such techniques and methods to make the most out of the educational data such as the different available Educational Resources (ERs) (Romero and Ventura, 2013).

Usually, ERs are considered to be text documents whether they contain text such as books, articles, and presentation slides or they can be transformed into text which is the case for the use of videos by using their transcripts. However, there are other differences that should also be taken into account such as the differences in sizes, languages, topics, levels, and concepts covered. This motivates the need for having a common representation that allows systems to process different ERs efficiently.

Let us assume that we have a set of ERs that differ in many ways (Figure 1). In order to be able to manipulate these ERs, we need to have a common representation for them. Otherwise, we will need to treat different ERs in different ways following their sizes, languages, formats, etc. One way to do that is to project all the ERs into a latent space so that every

[a] https://orcid.org/0009-0004-5209-6494
[b] https://orcid.org/0009-0002-1048-8663
[c] https://orcid.org/0000-0002-7399-0012
[d] https://orcid.org/0000-0002-3701-6263

ER has a vector representation, this process is known as embedding. In Figure 1, we use similarity embedding in which similar ERs are projected close to one another[1]. Since ERs $R_1$ and $R_2$ are covering the same subject, they are projected close to one another despite the differences in size, format and author. As for $R_3$, it is projected further than the first two as it covers a completely different subject despite the similarity in the release year, format and author with $R_2$ :

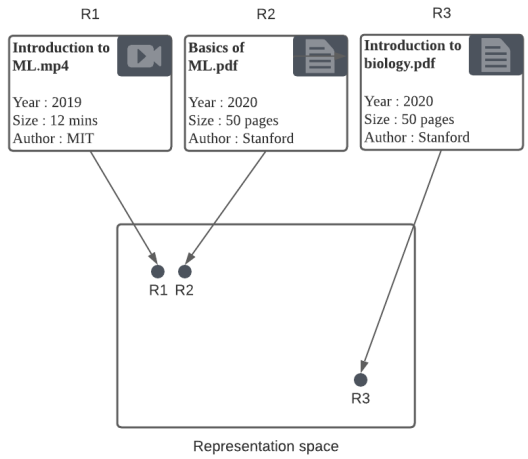


Figure 1: Embedding of educational resources.

Since ERs are considered to be text documents, researchers showed great interest in techniques based on document representation methods when manipulating them. Despite the remarkable results achieved using these methods, there are still some particularities to be considered when applying them to ERs. In this work, we are interested in mitigating two problems that prevent the generic text representation techniques from being directly used on ERs. First, these models are mostly used in tasks dependent on the writing style rather than the content (Mukherjee, 2021). Whereas for ERs, in most cases, the content is the most important part of the document. The content can be seen as the core of the ER. We want two ERs that cover the same concepts and discuss the same subject to have representations that are similar. However, for some of the usual methods, if an author writes two articles about different subjects, the representations generated for these two different articles would still have some sort of similarity because these representations take into considerations the entirety of the text, the writing style included. Second, the state of the art techniques seem to not generate representations that reflect a semantic representation of the text (Merrill et al., 2021). However, in ERs, we would like to have representations that are grounded on some sort of semantic structure.

In this article, we present a novel method to generate task-agnostic embeddings for ERs using document annotations and Linked Open Data. To explain our contribution, we start by discussing the related work and some preliminary notions in Section 2. Then, we discuss the architecture of the model used in Section 3. In Section 4, we conduct empirical tests on our method and prove its efficiency as it achieves better results than the state of the art methods. We conclude in Section 5 by summarizing the contribution and discussing some perspectives for future work.

# 2 RELATED WORK

## 2.1 State of the art

Data mining, and text mining more specifically, have been widely studied in education (Ferreira-Mello et al., 2019). Many techniques were used in education such as text classification (Lin et al., 2009), clustering (Khribi et al., 2008) and many more. These techniques were used on different texts related to education, namely ERs, forums, online assignment, etc. Text mining is used in many tasks related to education, it can be automatic evaluation of students (Crossley et al., 2015), recommender systems (Drachsler et al., 2015), program embedding learning (Cleuziou and Flouvat, 2021), etc. However, (Ferreira-Mello et al., 2019) reported that from the articles he analyzed in his survey, less than 3% used text mining techniques on documents (or what we have been referring to as ERs).

Outside of education, document representation techniques have been evolving at an incredible rate (Liu et al., 2020). Education can benefit from these breakthroughs and use these techniques to represent ERs. From these techniques, we mention the bag-of-words model, which is a one-hot document encoding method. TF-IDF is also widely used, it quantifies importance of the terms in a document. Topic Models such as LDA (Blei et al., 2003) is another method, but this technique identifies a list of topics that a text covers which is different from what we want to accomplish. Doc2Vec (Le and Mikolov, 2014), which is a generalization of Word2Vec, is another technique that is still being used. It is based on the use of the skip-gram model. Most recently, the use of large language models is getting more and more interest, especially the pretrained ones, such as BERT (Devlin

[1]We opted for a 2D representation for simplicity, the commonly used embeddings have bigger dimensions.

et al., 2019) which is a model based on a bidirectional deep transformer architecture.

The representations used for ERs and text documents play a major role in determining the quality of the results returned for any technique, whether it is classification, clustering, etc. Having an ER representation of quality is important when manipulating it. However, sometimes we do not have an objective function that can be optimized for the task at hand to generate suitable representations like some recommender systems. Furthermore, it can be interesting to have a representation that is task-agnostic and can be used as a building block for a more complex system.

Recommender systems in education have explored different ways to represent ERs (Urdaneta-Ponte et al., 2021). Traditionally, recommender systems are divided into three categories (Adomavicius and Tuzhilin, 2005). The first one is content-based (CB), the second is collaborative filtering (CF) and the third and final one is hybrid recommender systems. Since we are interested in the ERs' content, we will be focusing in the next paragraph on CB and hybrid recommender systems in education as they are the ones that focus on the ERs' representations. Even though we are focusing on recommender systems for ERs and that our method can be used in such systems, they are not the topic of this article as we are interested in ER representations in general.

To represent ERs in a recommender system, (Broisin et al., 2010) have used WordNet-based keyword processing combined with matrix decomposition. (Zhao et al., 2018) used a hybrid similarity measure based on link and object similarity where objects (ERs) were similar based on a comparison of their top-k TF-IDF terms. Other works such as (Lessa and Brandão, 2018) have used pipelines that transform the text into a vector by doing some preprocessing such as stop word removal, stemming and other techniques followed by the use of TF-IDF for weighting the filtered terms found in the text. (Ma et al., 2017) used a pipeline composed of some of the most used techniques in the text mining community (clean-tokenize-TFIDF-Word2Vec-Doc2Vec) to create an embedding for the courses that are used to make recommendations based on similarity. Other representations of ERs have been created outside of recommender systems. For example, (Li et al., 2022) have used BERT to create an embedding for ER descriptions and used them in a grade prediction model.

All of these representations suffer from at least one of the two aforementioned problems. For example, BERT is sensitive to the writing style. Other methods such as TF-IDF are not based on a semantic grounded representation. These problems, make the representations generated by these methods unsuitable for some tasks that require a special focus on the content such as recommender systems.

## 2.2 Preliminaries

Our contribution is in the form of a pipeline. In order to better present it, we first present two key concepts necessary to understand the architecture chosen for the pipeline.

- **Wikification :** A semantic annotation technique that uses Wikipedia as a source of possible semantic annotations by disambiguating natural language text and mapping mentions into canonical entities also known as Wikipedia concepts (Hoffart et al., 2011).
- **Knowledge base entity embedding :** Knowledge base entity embedding consists of training models to learn embeddings for entities in a knowledge base such as Wikipedia, DBPedia, Wikidata, etc (Hu et al., 2015).

# 3 EMBEDD-ER

Given the specificities of the ERs, our goal is to create a model that generates an embedding representative of the ER's content in a more compact format. In order to go from a text representation of an ER to an embedding, we create a pipeline composed of three components.

We start by taking the text of the ER and pass it to a Wikification tool known as Wikifier (Brank et al., 2017). Wikifier takes a text document as input (Table 1a) and annotates it with links to relevant Wikipedia concepts. As shown in Table 1b, this process is achieved through the identification of the different terms found in the input text, disambiguating them based on the context, linking them to a Wikipedia page, then attributing an importance score to every concept. To calculate this importance score, Wikifier[2] proceeds by creating a bipartite graph called the mention-concept graph. This graph contains the mentions found in the text in one set and the Wikipedia concepts in the other set. This graph is augmented by adding links between concepts using a semantic relatedness measure. The pageRank algorithm is then applied to the resulting graph, and the pageRank scores for the concepts are the importance scores for the concepts returned (Brank et al., 2017). The result of this step is a list of tuples composed of the concept found, its Wikipedia page, and its pageRank score. This first

[2]https://wikifier.org/info.html

Table 1: Generating embeddings with EMBEDD-ER on an ER text example.

(a) Input ER.

| |
|---|
| ... This is really what I mean by **race** being a very unexpected undercurrent in **The Great Gatsby**. This really is a completely unnecessary detail. We'll never see that **limousine** ... |

(b) Wikifier results.

| **Term** | **Wikipedia Link** | **PageRank** |
|---|---|---|
| Gatsby | http://en.wiki.../Gatsby | 0.0849 |
| Race | http://en.wiki.../Race | 0.0328 |
| Limousine | http://en.wiki.../Limousine | 0.0077 |
| ... | ... | ... |

(c) Wikipedia2Vec embeddings.

| **Embedding** | **PageRank** |
|---|---|
| $e_1 = [0.1...0.6]$ | 0.0849 |
| $e_2 = [0.2...0.6]$ | 0.0328 |
| $e_3 = [0.2...0.5]$ | 0.0077 |
| ... | ... |

step allows us to focus on the content and concepts covered in the ER instead of the writing style and other content-irrelevant details.

The result of the previous step is then used as the input to the following component which is a pretrained Knowledge base entity embedding model named Wikipedia2Vec (Yamada et al., 2020). Wikipedia2Vec is based on an extension of the skip-gram model (Yamada et al., 2016). It is trained on the paragraphs of the Wikipedia articles as well as a Wikipedia graph in which the nodes are Wikipedia entities and the edges represent the presence of hyperlinks between these entities, this Wikipedia graph is a subgraph of DBPedia. The Wikipedia2Vec model generates embeddings of size 300 for entities in Wikipedia. The use of Wikipedia2Vec has many advantages over using a model that is trained on the current data only, as TF-IDF. For instance, training a model on a knowledge base, such as Wikipedia, gives it a larger vision and a better representation of concepts and their relations. If two concepts are synonyms for example, Wikipedia2Vec would attribute similar embeddings to these two concepts, this means that if an ER uses more than one term to refer to the same concept, Wikipedia2Vec will be able to capture the similarity. Furthermore, being trained on Wikipedia's articles and graph gives the model a better semantic representation of subjects and concepts rather than only being trained on paragraph structures namely the popular models such as BERT that have been trained on the tasks of tokens masking and next sentence prediction (Devlin et al., 2019). We use a Wikipedia2Vec model, pretrained on a 2018 snapshot of Wikipedia, on the results returned by Wikifier to generate embeddings for the Wikipedia entities found in the text. Table 1c shows how for every Wikipedia concept, an embedding is associated. This step ensures the embeddings generated are grounded on Wikipedia, thus they vehicle a semantic meaning for the text concepts and the document as a whole. This simulates the notion of understanding the meaning of a concept.

Finally, we know that not every concept mentioned has the same importance within an ER. Therefore, we aggregate the concepts using a normalized weighted sum for every concept's embedding and its pageRank value to give more weight to the more important entities. The final result is an embedding for the input ER that is representative of its content with respect to a grounded understanding. To calculate an embedding $e$ for an ER that has $n$ concepts each having an embedding $e_i$ and a pageRank $p_i$, we use the following formula :

$$e = \frac{\sum_{i=1}^{n}(e_i \times p_i)}{\sum_{i=1}^{n} p_i} \quad (1)$$

To illustrate how the aggregation works, we generated an embedding of an ER from a Yale course (Figure 2). In this ER, the teacher discusses race in the novel of The Great Gatsby by F. Scott Fitzgerald. The embedding generated for the resource following the equation 1 (in red) is closer to the key concepts of the ER. However, the embedding generated by computing an unweighted average (in orange) seems less precise as it does not reflect the importance of the key concepts in the result embedding. T-SNE was used to reduce the dimensionality of the generated embeddings from 300 to 2 in order to plot the results.

# 4 EXPERIMENTAL EVALUATION

In order to verify if these embeddings are truly representative of these ERs, we decided to make two types of tests. First, we study the similarity of ERs through embeddings. Second, we use the embeddings generated in a multi-class subject classification[3].

The tests were conducted on a machine equipped with a 64 bit Fedora Linux 35 OS with 16 GB of RAM and an 11th Gen Intel i7-11850H @ 2.50GHz × 16 processor.

[3]https://gitlab.inria.fr/abazouzi/embedd-er

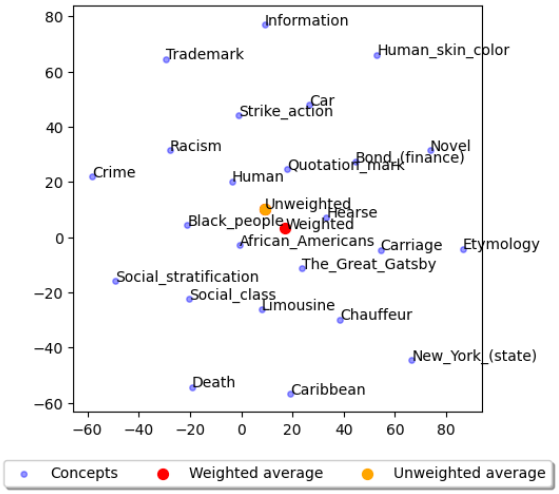


Figure 2: Aggregating the entities' embeddings.

## 4.1 Data

In these past years, a lot of efforts were made to make educational resources easily accessible and free to use for both students and educators. Resources of this type were named Open Educational Resources (OERs) by UNESCO. In this work, we are interested in using this type of ERs as they allow educators and learners to retain, reuse, revise, remix, and redistribute them.

We have used a dataset extracted from *Open Yale Courseware* (OYC)[4] (Connes et al., 2021). This dataset is formed in a hierarchical manner. It is composed of series, within which we find episodes, which are divided into chapters themselves. A chapter is the atomic building block of this dataset, it is represented as a text that covers a specific subject. We consider these chapters to be ERs. This dataset has in total 2550 chapters (ERs). However, we will not be using all of them due to the class imbalance between subjects for these ERs.

To measure similarity, we randomly pick four subjects : African-American studies, Biomedical Engineering, Economics, as well as Geology and geophysics. Then, from these subjects we chose 3 ERs per subject. This is an arbitrary choice and has been made to have a compromise between visual simplicity and representativeness. We use similarity measures to compare these 12 ERs. We also use the ERs belonging to the 3 most represented subjects in the dataset and use them both to test similarity in a visual manner and in classification tasks. The 3 subjects are : Ecology and Evolutionary Biology, African-American studies, and History. They have 407 ERs, 224 ERs, and 203 ERs respectively.

[4]https://oyc.yale.edu/

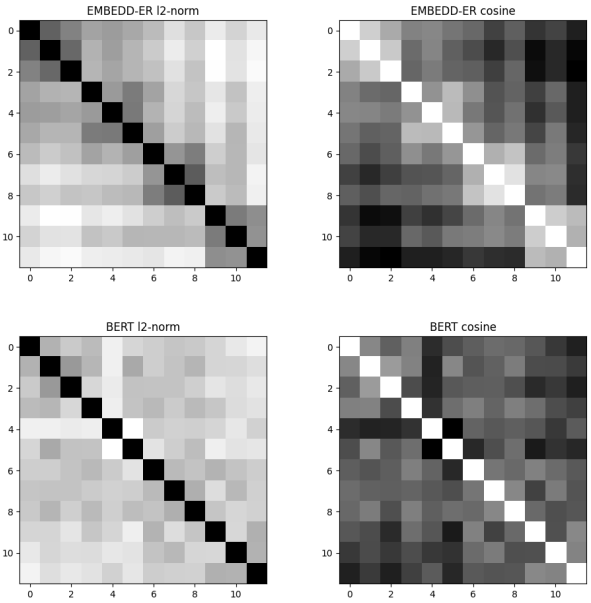


Figure 3: Similarity between embeddings generated by EMBEDD-ER and BERT for 12 ERs using the l2-norm and cosine similarity.

## 4.2 Results

### 4.2.1 Similarity

For testing the similarity, we use two measures that are widely used in embeddings comparison : the l2-norm and cosine similarity. We apply these measures to embeddings generated with EMBEDD-ER and BERT for 12 ERs from the OYC dataset, every 3 ERs belong to a different subject. The results are reported using matrix heat-maps.

In the similarity measures computed on the EMBEDD-ER embeddings (Figure 3), we notice that the resources belonging to the same subject are more similar to one another than to ERs of different subjects. This can be noticed by observing the $3 \times 3$ tiles formed along the diagonal axis of the similarity matrices. However, for the embeddings generated by BERT, these similarities are not as visible. This means that the embeddings generated by our method, project similar ERs close to one another and relatively far from different ERs. This can be explained by the fact that our method takes only the concepts which is not the case for BERT.

To further investigate similarity, we performed dimensionality reduction on the ERs belonging to the 3 most represented subjects. We reduced the dimensionality using T-SNE from d=300 for EMBEDD-ER and d=768 for BERT to d=2.

From the results presented in Figure 4, we can observe that the generated 2D coordinates for

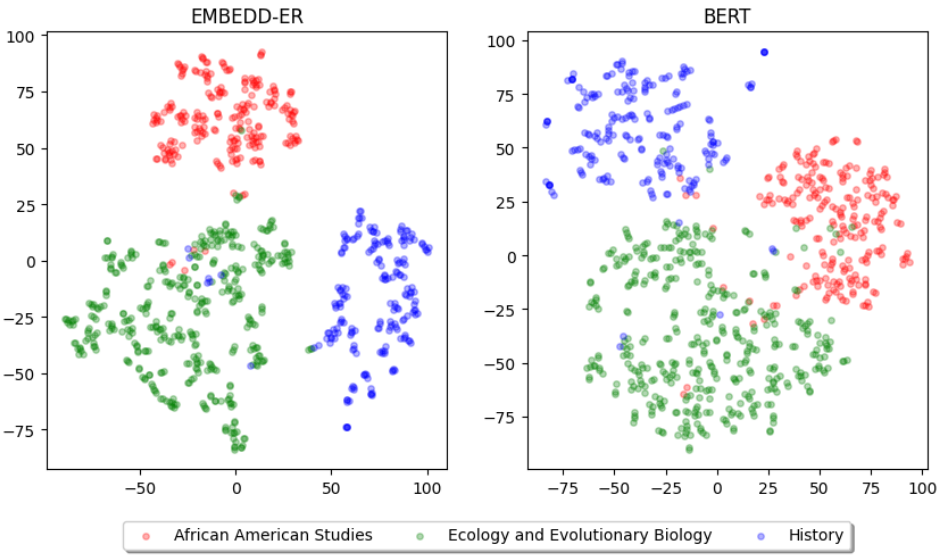


Figure 4: T-SNE performed on the embeddings generated by BERT and EMBEDD-ER for ERs covering 3 different subjects.

EMBEDD-ER have formed homogeneous clusters, in which ERs covering the same subjects (in the same color) are closer to one another than to other ERs. However, the coordinates generated for the BERT embeddings are more dispersed with different ERs projected close to one another than to similar ERs in some cases.

#### 4.2.2 Classification

To test our method, we used embeddings of the ERs belonging to the three most present subjects in the *OYC* dataset mentioned in Section 4.1.

For the baselines, we used the following models that have been mentioned in the related work section :

- BERT : We used a pretrained BERT model to generate embeddings of size 768.
- TF-IDF : We applied TF-IDF on the ERs of the three classes which generated embeddings of size 18508. An embedding of this size is too big. Therefore, we applied T-SNE to reduce the size of the embeddings to 700 so that it is close to the size of the BERT embeddings.
- Doc2Vec : We created a pipeline that removes stop words, does lemmatization then trains a Doc2Vec model on the processed text. The generated embeddings have the same size as TF-IDF (700).

We used a 4-fold cross-validation grid search to determine the best hyper-parameters for 4 classification models : Decision trees, SVM, Naive Bayes (GNB), and KNN. The results reported represent the mean accuracy of a 4-fold cross validation for these models with the best hyper-parameters found with grid search.

Table 2: Comparison of multi-class classification accuracy between TF-IDF, Doc2Vec, BERT, and EMBEDD-ER.

| Method | Decision tree | SVM | GNB | KNN |
|---|---|---|---|---|
| TF-IDF | 76.97% ±6.32% | 65.81% ±8.38% | 78.27% ±11.3% | 85.23% ±9.18% |
| Doc2Vec | 48.56% ±0.51% | 48.8% ±0.17% | 44.12% ±1.79% | 44.61% ±2.91% |
| BERT | 78.41% ±1.73% | 94.12% ±3.13% | 90.52% ±5.48% | 92.44% ±5.61% |
| EMBEDD-ER | **91.24% ±3.55%** | **98.2% ±0.71%** | **95.8% ±4.87%** | **97.36% ±1.1%** |

In multi-class classification, EMBEDD-ER is the best model followed by BERT (Table 2). These results highlight the fact that despite not having trained EMBEDD-ER on classification tasks, it still performs better than the state of the art models used for document representations while generating more compact representations.

Table 3 summarizes for each model the different steps and the time necessary. For EMBEDD-ER, it takes a lot of time annotating the input ERs. However, we notice that EMBEDD-ER and BERT generate the embeddings faster than the TF-IDF and Docvec with a slight advantage for EMBEDD-ER. Furthermore, BERT is a pretrained model and EMBEDD-ER uses Wikipedia2Vec which is also a pretrained model which means that they need time to be loaded

Table 3: Comparison of the time consumed (in seconds) for the different steps of TF-IDF, Doc2Vec, BERT, and EMBEDD-ER.

| Method | Pre-processing | Model loading | Embedding |
|---|---|---|---|
| TF-IDF | × | × | 635.80 |
| Doc2Vec | 4.15 | × | 705.28 |
| BERT | × | 2.15 | 217.72 |
| EMBEDD-ER | 5695.89 | 517.19 | 142.32 |

into memory. Since EMBEDD-ER takes more time to be loaded but less time to generate embeddings compared to BERT, it means that for big preprocessed (annotated with Wikifier) datasets EMBEDD-ER can be faster than BERT.

#### 4.2.3 Ablation study

Given that EMBEDD-ER is made by combining three components, it can be interesting to alter its architecture and check the performance of the different generated versions and study the impact of altering the components. For that purpose, we have created three more versions of EMBEDD-ER :

- BASE : The model presented in Section 3.
- BASE UW : Same as BASE, but for the aggregation is computed using an unweighted average :

$$e = \frac{\sum_{i=1}^{n} e_i}{n} \tag{2}$$

- BERT-ER : Same as BASE, but we use BERT instead of Wikipedia2Vec for generating the concepts' embeddings.
- BERT-ER UW : Same as the previous model (BERT-ER) but we aggregate the embeddings using Equation 2's formula.

We use the different architectures in a multi-class classification task. Using the same experimental setup presented in the previous section. The results are reported below :

By observing the results of the multi-class classification reported in Table 4, we notice that BASE outperforms all the models when used with Decision trees, GNB and KNNs while BASE UW performs better with SVM and gets the second best results with the remaining classifiers.

As a general conclusion for the ablation study, the most important part of the architecture is the use of Wikipedia2Vec as the top two performing models use it. The use of Wikipedia2Vec allows the embeddings to vehicle semantic meaning and the use of a weighted average adds more precision.

Table 4: Ablation study on multi-class classification accuracy.

| Method | Decision tree | SVM | GNB | KNN |
|---|---|---|---|---|
| BASE | **91.24% ±3.55%** | 98.20% ±0.71% | **95.8% ±4.87%** | **97.36% ±1.1%** |
| BASE UW | 89.68% ±3.65% | **98.44% ±0.71%** | 94.23% ±6.43% | 97.12% ±0.59% |
| BERT-ER | 81.05% ±5.06% | 94.96% ±1.25% | 74.1% ±1.51% | 84.77% ±2.58% |
| BERT-ER UW | 76.50% ±1.62% | 96.29% ±1.48% | 74.59% ±3.74% | 79.62% ±2.44% |

## 5 CONCLUSIONS

In this work, we presented a new method for generating homogeneous, compact, writing-style-independent, and content-focused representation for ERs. This method is based on the use of a pipeline that generates an embedding from an ER's text by using document annotations and a knowledge base embedding model pretrained on Wikipedia.

The embeddings generated by this method have been tested in two different ways. The first was a similarity test, it showed that embeddings of ERs covering the same subjects are projected close to one another in the embedding space. The second was a subject classification task in which EMBEDD-ER performed better than the state of the art text representations. This is due to EMBEDD-ER generating embeddings using the content while taking into account the importance scores of the different concepts covered by the ER.

As for future works, we want to test EMBEDD-ER on other datasets of different sizes, languages, and subjects to test its generalization capabilities. It can also be interesting to have a local implementation of Wikifier to avoid using the web API or even test other tools for annotating the input text. This might reduce the computation time for our method. We would also like to use this method in real world tasks such as incorporating it in a recommender system, especially content-based ones, or in an automatic curriculum design task.

## ACKNOWLEDGEMENTS

This work has received a French government support granted to the Labex Cominlabs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01.

# REFERENCES


Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Brank, J., Leban, G., and Grobelnik, M. (2017). Annotating documents with relevant Wikipedia concepts. *Proceedings of SiKDD*, 472.

Broisin, J., Brut, M., Butoianu, V., Sedes, F., and Vidal, P. (2010). A personalized recommendation framework based on cam and document annotations. *Procedia Computer Science*, 1(2):2839–2848.

Cleuziou, G. and Flouvat, F. (2021). Learning student program embeddings using abstract execution traces. In *14th International Conference on Educational Data Mining*, page 252–262.

Connes, V., de la Higuera, C., and Le Capitaine, H. (2021). What should I learn next? ranking educational resources. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, page 109–114.

Crossley, S., Allen, L. K., Snow, E. L., and McNamara, D. S. (2015). Pssst... textual features... there is more to automatic essay scoring than just you! In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, LAK '15, page 203–207, New York, NY, USA. Association for Computing Machinery.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. (arXiv:1810.04805). arXiv:1810.04805 [cs].

Drachsler, H., Verbert, K., Santos, O. C., and Manouselis, N. (2015). *Panorama of recommender systems to support learning*, page 421–451. Springer.

Ferreira-Mello, R., André, M., Pinheiro, A., Costa, E., and Romero, C. (2019). Text mining in education. *WIREs Data Mining and Knowledge Discovery*, 9(6):e1332.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, page 782–792.

Hu, Z., Huang, P., Deng, Y., Gao, Y., and Xing, E. (2015). Entity hierarchy embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, page 1292–1300.

Khribi, M. K., Jemni, M., and Nasraoui, O. (2008). Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. In *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, page 241–245.

Kirkwood, A. and Price, L. (2014). Technology-enhanced learning and teaching in higher education: what is 'enhanced' and how do we know? a critical literature review. *Learning, media and technology*, 39(1):6–36.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, page 1188–1196. PMLR.

Lessa, L. F. and Brandão, W. C. (2018). Filtering graduate courses based on LinkedIn profiles. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, page 141–147.

Li, J., Supraja, S., Qiu, W., and Khong, A. W. (2022). Grade prediction via prior grades and text mining on course descriptions: Course outlines and intended learning outcomes. *International Educational Data Mining Society*.

Lin, F.-R., Hsieh, L.-S., and Chuang, F.-T. (2009). Discovering genres of online discussion threads via text mining. *Computers Education*, 52(2):481–495.

Liu, Z., Lin, Y., and Sun, M. (2020). *Representation learning for natural language processing*. Springer Nature.

Ma, H., Wang, X., Hou, J., and Lu, Y. (2017). Course recommendation based on semantic similarity analysis. In *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)*, page 638–641. IEEE.

Merrill, W., Goldberg, Y., Schwartz, R., and Smith, N. A. (2021). Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? *Transactions of the Association for Computational Linguistics*, 9:1047–1060.

Mukherjee, S. (2021). *Sentiment Analysis*, page 113–127. Apress, Berkeley, CA.

Romero, C. and Ventura, S. (2013). Data mining in education. *WIREs Data Mining and Knowledge Discovery*, 3(1):12–27.

Urdaneta-Ponte, M. C., Mendez-Zorrilla, A., and Oleagordia-Ruiz, I. (2021). Recommendation systems for education: Systematic review. *Electronics*, 10(1414):1611.

Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., and Matsumoto, Y. (2020). Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia. (arXiv:1812.06280). arXiv:1812.06280 [cs].

Yamada, I., Shindo, H., Takeda, H., and Takefuji, Y. (2016). Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*.

Zhao, Q., Wang, C., Wang, P., Zhou, M., and Jiang, C. (2018). A novel method on information recommendation via hybrid similarity. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(3):448–459.