Improving Embeddings Representations for Comparing Higher Education Curricula: A Use Case in Computing

Anonymous ACL submission

Abstract

We propose an approach for comparing curricula of study programs in higher education. Pre-trained word embeddings are fine-tuned in a study program classification task, where each curriculum is represented by the names and content of its courses. By combining metric learning with a novel course-guided attention mechanism, our method obtains more accurate curriculum representations than strong baselines. Experiments on a new dataset with curricula of computing programs demonstrate the intuitive power of our approach via attention weights, topic modeling, and embeddings visualizations. We also present a use case comparing computing curricula from USA and Latin America to showcase the capabilities of our improved embeddings representations.

1 Introduction

Several stakeholders in Education compare and assess curricula, such as Governments aiming to improve the competitiveness of their local programs (Adamson and Morris, 2007), or academics looking to analyze market offer and suggest new courses and innovations (Prietch, 2010; de Alburquerque et al., 2010; Macêdo, 2016). Despite their benefits, manual comparisons can be time-consuming, and are prone to biases from human assessors due to their personal beliefs and perspectives (Kawintiranon et al., 2016). This situation becomes more challenging when studying curricula related to programs in computing due to their fast-paced evolution, which may prevent suitable analysis and evaluation by human intervention (Föll and Thiesse, 2021). To aid higher education stakeholders, we propose a method to automatically compare curricula based on the names and content of their courses, and test it on study programs in computing.

Previous work has represented curricula using TF-IDF, and used clustering algorithms to identify groups of commonly-studied topics in data science (West, 2017), or to find links between study programs in different countries and across computing disciplines (Murrugarra-Llerena et al., 2011). Others have attempted to associate curricula to knowledge areas defined by international associations, such as ACM or IEEE (Shackelford et al., 2005), using standard TF-IDF representations (Kawintiranon et al., 2016) or topic modelling (Matsuda et al., 2018; Föll and Thiesse, 2021).

In this work, we follow the current trend in NLP applications and use pre-trained word embeddings, such as BERT (Devlin et al., 2019), to obtain better representations of textual curricula. We fine-tune these embeddings on a computing discipline classification task, using a newly-collected dataset that includes study programs from accredited universities in the USA and LATAM. We introduce a course-guided attention mechanism that allows the model to identify which courses are more relevant for each computing discipline, and pair it with metric learning (Xing et al., 2002) to learn well-defined groups. Experiments with different types of pre-trained word embeddings and classification algorithms, show that our proposed approach allows an accurate representation to separate computing curriculums and allow human understanding. We also show qualitative results via attention weights, topic modeling, and embedding visualizations. These results highlight the performance of our approach by identifying relevant words for each computing curriculum. Finally, we develop an application to visualize how Latin America computing programs are identified with international ABET careers.

In summary, our main contributions are:

• A novel dataset of US computing curricula, with relevant Latin America universities.
• An examination of attention, metric learning, and Bert modules to generate more intuitive embedding representations.
• An application to categorize a computing curriculum compared to international standards.
2 Computing Curricula Dataset

We collected curricula from university study programs from different countries and for five computing disciplines: Computer Science (CS), Computer Engineering (CE), Information Technology (IT), Information Science (IS), and Software Engineering (SE). Each curriculum consists of a set of courses including their title and course description (see examples in Table 3 in Appendix A). The dataset consists of two sections:

- **USA.** Contains 296 curricula from US universities in the top 1000 of the QS rankings from 2021, and that were also accredited by ABET.
- **LATAM.** Contains 18 curricula from high-ranking universities in Brazil, Colombia, Mexico, and Peru. These study programs claim to correspond to the Computer Science discipline. These curricula were first translated using Google Translate, and then manually revised to resolve consistency issues such as mistranslated words, wrong word order, etc.

The USA portion of the dataset is used to train and evaluate embeddings representation in a computing discipline classification task (Sec. 5.1), while the LATAM portion is exploited to analyse their importance of each course associated with computing curriculums. As shown in Figure 1, it receives a computing curriculum composed of courses and their *Bert* embeddings $\text{curriculum}_{emb}$. Then, we compute $\text{att}_i$ weights of each course. Using these weights, we calculated a weighted average over the courses and generate a new curriculum embedding $\text{curriculum}_{emb\_avg}$. Finally, we collapsed the generated embedding in 128 features.

**Metric Learning.** Using these features, we learn well-defined groups among computing curriculums. We employ metric learning with the following triplet loss (Schroff et al., 2015), where $N$ is the number of instances in a batch, $\alpha$ is the triplet margin with value 0.3, and $\theta$ denotes the learnt parameters.

$$ L(c; \theta) = \sum_{i=1}^{N} \left[ \frac{1}{2} ||c_i|| \times ||c_a^p|| - ||c_i|| \times ||c_a^n|| + \alpha \right] $$

Given an anchor curriculum ($c_a^p$) and using instances in the same batch, we select curriculums with the same category as positive annotations ($c_a^p$), and curriculums from different categories as negative annotations ($c_a^n$). Data points were randomly sampled with a batch size of 64 to ensure that every category is present in each iteration.

3 Approach

In order to obtain better representations of textual curricula, we propose to use pre-trained BERT (Devlin et al., 2019) embeddings that have been fine-tuned in a computing discipline classification task, using an approach that combines a novel course-based attention mechanism and metric learning. Figure 1 shows an overview of our method. Course-based attention identifies more and less important courses following the intuition of core and elective courses, while metric learning learns boundaries to form well-defined groups.

**Course-Based Attention.** Our course-based attention approach $\text{Bert}_{\text{met} + \text{att}}$ aims to learn the importance of each course associated with computing curriculums. As shown in Figure 1, it receives a computing curriculum composed of courses and their $\text{Bert}$ embeddings $\text{curriculum}_{emb}$. Then, we compute $\text{att}_i$ weights of each course. Using these weights, we calculated a weighted average over the courses and generate a new curriculum embedding $\text{curriculum}_{emb\_avg}$. Finally, we collapsed the generated embedding in 128 features.

![Figure 1: Our course-based attention approach. It generates an intuitive representation of curriculums via attention weights and metric learning. Attention highlights core courses, while metric learning learns boundaries to form well-defined groups. Both components are crucial to find accurate representations.](image)

**Metric Learning.** Using these features, we learn well-defined groups among computing curriculums. We employ metric learning with the following triplet loss (Schroff et al., 2015), where $N$ is the number of instances in a batch, $\alpha$ is the triplet margin with value 0.3, and $\theta$ denotes the learnt parameters.

$$ L(c; \theta) = \sum_{i=1}^{N} \left[ \frac{1}{2} ||c_i|| \times ||c_a^p|| - ||c_i|| \times ||c_a^n|| + \alpha \right] $$

Given an anchor curriculum ($c_a^p$) and using instances in the same batch, we select curriculums with the same category as positive annotations ($c_a^p$), and curriculums from different categories as negative annotations ($c_a^n$). Data points were randomly sampled with a batch size of 64 to ensure that every category is present in each iteration.

3.1 Implementation details

We implemented the networks using PyTorch (Paszke et al., 2019) and metric learning library (Musgrave et al., 2020) on a RTX 3060 GPU. We ran each experiment ten times with different seeds using SGD. From preliminary experiments, we vary the batch size in the range $[32, 64, 128]$ and the embedding output

---

2. [https://www.abet.org/](https://www.abet.org/)
3. scored by Google search
4. for research purposes

---
in the range $[128, 256, 512]$. Then, we select the best configuration (batch size = 64 and embedding size = 128) in our validation set.

### 4 Experimental Setting

#### 4.1 Baselines

We compare six traditional methods: Majority, Random, TF-IDF (Sammut and Webb, 2010), Word2vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), and Bert (Devlin et al., 2019)

Additionally, we fine-tuned BERT’s embedding with our training data:

- $\text{Bert}_{\text{unsup}}$: unsupervised finetuning using language modeling.
- $\text{Bert}_{\text{sup}}$: supervised finetuning using labels.

We also consider metric learning competitors:

- $\text{Bert}_{\text{met}}$: supervised metric learning using Bert.
- $\text{Fusion}_{\text{met+att}}$: supervised metric learning with attention over Glove, Word2vec and Bert.

#### 4.2 Evaluation protocol

From our USA dataset, we split our data in 60% for training, 20% for validation, and 20% for testing in an stratified fashion. Using the train set, in non-traditional methods\(^1\), we learn a new embedding representation. Then, we use those embeddings to feed machine learning classifiers. To select the best parameter configuration, each classifier was evaluated on a validation set and the configuration with higher F1 was selected for testing. For all non-pretrained models, we trained them with ten different seeds and report their F1 average.

### 5 Results and Analysis

#### 5.1 Quantitative experiments

We aim to validate which approach generates a more precise representation for classification, and human intuition. From the computed embeddings, we trained four classifiers: K-nearest neighbour (KNN), Linear Regression (LR), Linear Support Vector Machine (LSVM), and Radial Support Vector Machine (RSVM) with a proper search range of parameters (detailed in Section A.2.1).

Table 1 shows F1-score for all the embeddings. Each approach embedding size are reported on the last column. We observe that $\text{Bert}_{\text{met+att}}$ outperforms on average all other competitors and boosts the RSVM classifier. This finding suggest that generated embeddings differentiate better computing curriculums, and can be helpful for visualization tasks. On the other hand, $\text{Fusion}_{\text{met+att}}$ is the second-best performer and reports competitive results with the KNN and RSVM classifiers. From pre-trained embeddings, the best baseline is $\text{Bert}$ and presents the best result in LR and LSVM. On the other hand, the weakest baselines are Majority and Random.

To conclude, we believe our improved performance is due to our intuitive embedding via attention weights and metric learning modules.

#### 5.2 Qualitative experiments

##### 5.2.1 Embedding visualizations

To understand if our approach generates meaningful embeddings, we visualize Bert and $\text{Bert}_{\text{met+att}}$ through Umap (McInnes et al., 2018) in Fig. 2.

$\text{Bert}_{\text{met+att}}$ separates more clearly computing programs than Bert. CE and CS show well-defined boundaries rather than in Bert Umap, and overlap is minimized among all categories. We also observe that IT and IS are close by. A possible explanation is through their shared financial and administration components. On the other hand, SE seems to be hard to form its own group. Apparently, it has pieces of all careers. We attribute this finding due that SE is a new career, less well-established. Finally, we also analyze the attention weights of our best competitor $\text{Fusion}_{\text{met+att}}$ in Section A.3.4. Bert is the most important representation, which confirms our choice of Bert embedding.

##### 5.2.2 Attention weights

To analyze the internal functionality of our approach and to verify if our model identifies core
courses, from the $\text{Bert}_{\text{met+att}}$ model, we extract the attention weights of each computing course. Then, we rank them in decreasing order and select the top five. We group these selected courses per computing program and create a word cloud visualization using course titles.

Figure 3 shows these computed word clouds for each computing program. We find that words with a higher number of occurrences are relevant to their respective category name. We observe that “computer” is common among all computing careers, but it is more relevant for CS, CE, and SE; while it has less importance for IT and IS.

CS suggests a strong association to algorithms and computer; CE to design and computer. IT to Information Management and System; IS to Principles and Information Database and SE to Systems and Programming. All these associations confirm the identity of each career, and we observe that IT and IS highlight information-related courses, while CS, CE, and SE are more technical. For example, CS focuses on algorithm efficiency, CE specializes in hardware design, and SE promotes programming skills in general. The frequencies of each word by category are in Section A.3.2, and also a comparative analysis of word frequency with TF-IDF is shown in Section A.3.3. Also, we selected most frequent course titles, and identify topics using (Popa and Rebedea, 2021)\(^8\) in Section A.3.1. Similarly, we observe key differences among careers.

6 Application: Internationalization

For our application, we investigate how LATAM computing careers relate to international standards. We used $\text{Bert}_{\text{met+att}}$ to project unseen CS LATAM computing curriculums and relate them with US standards in Figure 2 (c) using Umap.

LATAM curriculums (in triangles) form two groups: one near CS, and the other group surrounding IS and SE. Also, no LATAM curriculum surrounds the CE profile. From this visualization, we infer that LATAM curriculums are different from the US because none of them lay inside US groups. Then, we perform a closer study on individual LATAM countries. Brazil and Mexico have a clear CS profile. Also, Mexico seems much more integrated with the US profile. On the other hand, Peru has a mixed profile between CS, SE, and IS; which may suggest a better definition of courses per career. Finally, Colombia belongs to SE.

7 Conclusion

In this paper, we explore an intuitive way to generate precise representations for computing curricula understanding combining course-guided attention and metric learning. Our approach finds more cohesive groups with clear separation among them. These groupings are helpful for different machine learning models. Also, we analyze what our approach learns via attention weights, topic modeling, and visualization techniques.

\(^8\)https://huggingface.co/cristian-popap/bart-tl-all
References


A Appendix

We provide dataset statistics and additional details for quantitative and qualitative experiments. For quantitative, we provide details about parameter ranges for model selection. For qualitative experiments, we provide results on topic modeling, show counts for attended courses from our attention module, comparative word cloud analysis with TF-IDF, and attention weights for the best baseline competitor. Finally, we show limitations of our proposed approach.

A.1 Dataset Statistics

We compute statistics of our dataset such as number of curriculums, and average number of courses
per curriculum category with their standard deviations in Table 2. Also, as a way of visualization, we create a word cloud of all our dataset using course titles and their descriptions in Figure 4. We observe many courses are introductory, and applied; because words such as introduction and application are more frequent. On the other hand, core courses of computing categories can not easily found ensuring that our dataset is not biased to a specific class regardless of the imbalance dataset.

<table>
<thead>
<tr>
<th>Career</th>
<th>USA</th>
<th>LATAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>100</td>
<td>48.38±25.82 18 69.00±18.90</td>
</tr>
<tr>
<td>CE</td>
<td>98</td>
<td>53.71±22.10 - -</td>
</tr>
<tr>
<td>IT</td>
<td>37</td>
<td>43.10±16.91 - -</td>
</tr>
<tr>
<td>IS</td>
<td>34</td>
<td>40.38±15.60 - -</td>
</tr>
<tr>
<td>SE</td>
<td>27</td>
<td>46.25±13.62 - -</td>
</tr>
<tr>
<td>Total</td>
<td>296</td>
<td>49.67±33.69 18 69.00±18.90</td>
</tr>
</tbody>
</table>

Table 2: Statistics of our dataset. We enumerate number of curriculas and average number of courses per curriculum category and their associated standard deviations.

A.3 Qualitative experiments

A.3.1 Topic modeling

As a complementary way to understand our selected courses, we selected the ten course titles with highest attention, and input them to BART topic model (Popa and Rebedea, 2021) to name them.

The named topics are shown in Table 4. CS, CE, and IT share the word computer highlighting the importance of computing fundamentals, while IT and IS share the topics management and information relating to business knowledge. Also, programming skills are shared among CS and SE curriculums.

A.3.2 Counts for attended courses

Figure 6 shows the frequency of the top fifteen courses per category in decreasing order. We find the following associations per each computing career:

- CS highlights computer, introduction, system, design, algorithm, programming, and data courses.
- CE focuses on systems, design, computer, digital, and embedded.
- IT has relevant terms such as system, management, information, web, and programming.
- IS focuses on systems, principles, information, database, and management.
- SE highlights programming, systems, data, introduction, C, and software.

In summary, IT and IS are related to management and information knowledge. CE focuses on hardware concepts such as systems, design, and digital. Finally, CS and SE focus on software development related to programming, data, and algorithm courses.

For Linear SVM (LSVM), we evaluate cost C with values \([2^{-5}, 2^{-3}, 2^{-1}, 2^1, ..., 2^{15}]\).

For Radial SVM (RSVM), we evaluate cost C with values \([2^{-5}, 2^{-3}, 2^{-1}, 2^1, ..., 2^{15}]\), and gamma with values \([2^{-15}, 2^{-13}, 2^{-11}, ..., 2^1, 2^3}\).

A2 Quantitative experiments

A2.1 Range parameters for experiments

We mention the employed machine learning models with their associated parameter selection ranges below:

- For k-nearest neighbour (KNN), we evaluate k with values \([3, 5, 7]\).
- For Linear Regresion (LR), we evaluate cost C with values \([2^{-5}, 2^{-3}, 2^{-1}, 2^1, ..., 2^{15}]\).

9https://huggingface.co/cristian-popa/bart-tl-all
Career | Course title | Description
--- | --- | ---
CS | Algorithms and Data Structures | Study of data structures and algorithms ...
CE | Computer Architecture and Design | Principles of RISC-type CPU instruction set and ...
IT | Information Technology Security | Information technology security from a manager ...
IS | Information Systems Applications | Concepts and production skills ...
SE | Software Engineering Design | Techniques and methodologies ...

Table 3: Sample curriculum showing course titles and their description per computing career.

![Image](https://example.com/image1.png) ![Image](https://example.com/image2.png) ![Image](https://example.com/image3.png) ![Image](https://example.com/image4.png) ![Image](https://example.com/image5.png)

Figure 5: Word cloud comparison of $\text{Bert}_{\text{met+att}}$ and $\text{TF-IDF}$ among five computing careers.

<table>
<thead>
<tr>
<th>Career</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>computer programming data</td>
</tr>
<tr>
<td>CE</td>
<td>computer design system</td>
</tr>
<tr>
<td>IT</td>
<td>management information computer</td>
</tr>
<tr>
<td>IS</td>
<td>system management information data</td>
</tr>
<tr>
<td>SE</td>
<td>software programming language</td>
</tr>
</tbody>
</table>

Table 4: Topic identified with each computing curriculum using BART (Popa and Rebedea, 2021) model.

A.3.3 Word cloud TF-IDF vs Course-guided attention

We compare word clouds from $\text{TF-IDF}$ with our approach $\text{Bert}_{\text{met+att}}$ in Figure 5. We observe that $\text{TF-IDF}$ word clouds are more pollute with non-related words as opposite to our approach. Also, some important words such as: 
*programming, analysis, structure* are less predominant for CS in the TF-IDF wordcloud. A extreme case can be seen in SE, where is hard to identify predominant words for TF-IDF. In contrast, our approach identifies important words: 
*Software, programming, systems, data,* etc. For CE, our approach shows a strong relationship between 
*digital, computer, design* among others than $\text{TF-IDF}$. For IT and IS, TF-IDF does not show all important words such as: 
*management, information, web.* Finally, for all categories, $\text{TF-IDF}$ can identify relevant words, but also present meaningless ones such as 
*course, student, including,* among others.

A.3.4 Attention weights best competitor

We analyze our best competitor $\text{Fusion}_{\text{met+att}}$ to discover interesting knowledge. We extract attention weights for each embedding representation. On average, we obtained 0.2149 weight for $\text{Glove}$, 0.0621 for $\text{Word2vec}$, and 0.7230 for $\text{Bert}$. This finding confirms our election to select $\text{Bert}$ embedding in our approach. Also, it is interesting to see that $\text{Glove}$ and $\text{Word2vec}$ also have complementary and meaningful knowledge for better representation. Probably $\text{Word2vec}$ and $\text{Glove}$ provide local information to the $\text{Bert}$ embedding. Note, that their attention scores have the same order as their correspondent F1-score (see rows 4 to 6 in Table 1).

A.4 Limitations

Despite the good results of our approach, it also has some associated limitations to computational power, and associations of core course to computing careers

**Computational power.** As our model uses Bert, our model requires a forward pass of this deep model and GPU infrastructure for faster prediction times.

**Inability to associate a core course to a specific computing career:** Our model can identify core courses in general, however it miss to identify importance per computing career. For e.g. "Ad-
Figure 6: Top fifteen frequency terms of each category. The X-axis shows the word term from course titles, while Y-axis shows their frequency. The categories are (a) Computer Science (CS), (b) Computer Engineering (CE), (c) Information Technology (IT), (d) Information System (IS), and (e) Software Engineering (SE).

“Advanced Algorithms” is more important to a Computer Science curriculum but not to Information System. Unfortunately, our model is not able to distinguish that. In future work, we plan to develop an attention module per computing career.