Classification of educational videos by using a semi-supervised learning method on transcripts and keywords*

Alexandru Stefan Stoica[a,*], Stella Heras[b,*], Javier Palanca[b], Vicente Julián[b], Marian Cristian Mihaescu[a]

[a] *University of Craiova. Faculty of Automation, Computers and Electronics*
[b] *Universitat Politècnica de València. Valencian Research Institute for Artificial Intelligence*

Abstract

E-learning is a rapidly growing field, which is giving rise to a massive amount of digital learning objects. Sorting these objects properly so that they are correctly indexed in searches and recommendation systems is a challenge. In this paper, we present a semi-supervised method of clustering and classifying learning objects in video format to extract their most relevant topics, specifically from lesson transcripts. These videos come from the educational video platform of the *Universitat Politècnia de València*. The proposed method also uses open content from Wikipedia to help build the labelled dataset.

Keywords: language models, e-learning, classification

1. Introduction

Over the last few years, topic modelling has continued being a critical topic that affects different related areas such as machine learning (ML), natural language processing (NLP), information retrieval (IR) and other research communities. Among other proposals, *n-gram* statistical and probabilistic models [1, 2],

*Title Note
*Alexandru Stefan Stoica, Stella Heras

Email addresses: stoicaastefan@gmail.com (Alexandru Stefan Stoica), sheras@dsic.upv.es (Stella Heras), jpalanca@dsic.upv.es (Javier Palanca), vinglada@dsic.upv.es (Vicente Julián), mihaescu@software.ucv.ro (Marian Cristian Mihaescu), stoicaastefan@gmail.com, sheras@dsic.upv.es (Marian Cristian Mihaescu)

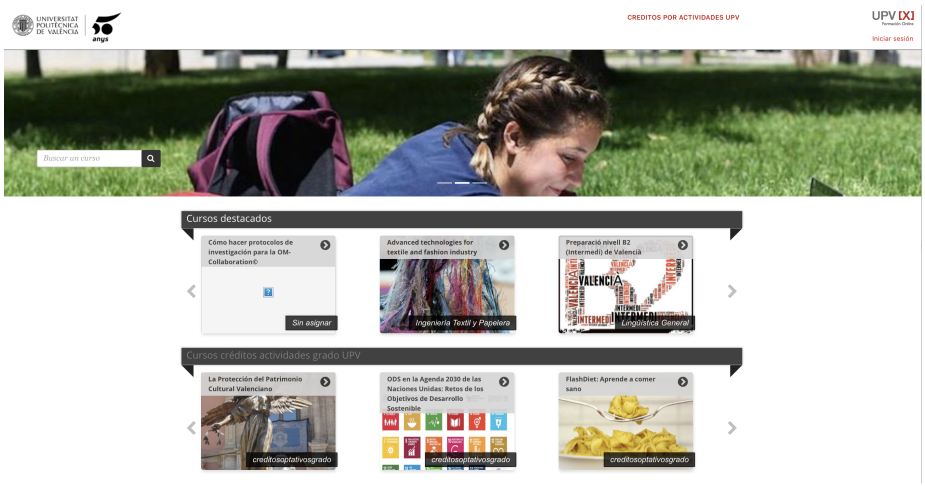

Figure 1: Website of the UPVx MOOC platform

hybrid approaches [3], and *bag-of-words* based models have been applied during these last years in order to discover topics and hidden semantic structures in text.

In parallel, online learning has experienced a considerable boom due mainly by the availability of an increasing quantity of online learning objects (LO) [4] and also, the emergence of Massive Online Learning Courses (MOOCs). To this must be added new teaching methodologies such as *Flipped Classrooms* [5] where the idea is to flip the common instructional approach: the teacher creates online LOs -videos and interactive lessons-, which students visualize at home to devote classes to work through problems and engage in collaborative learning). All this significant progress in online learning has led to the unwanted effect, the information overload problem. Due to this problem, students have more learning objects on the web than those who can locate and assimilate. According to this, topic modelling can be a desirable solution for current research in e-learning. The idea is to improve the searching process of objects that best-fit the typical keyword searches made by the students during their learning process.

The *Universitat Politècnica de València* (UPV, Spain) in line with this trend,

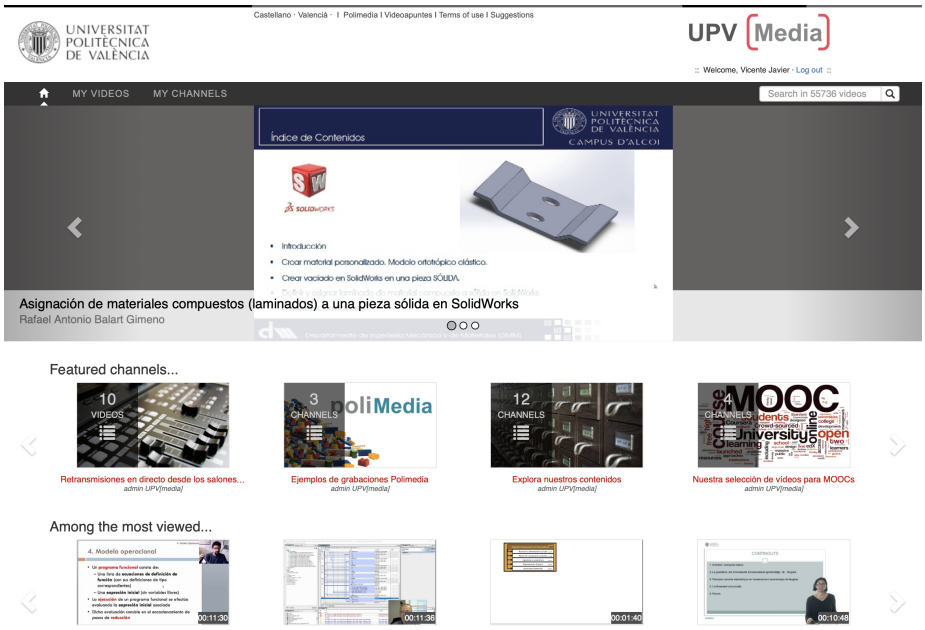

Figure 2: Website of the Universitat Politècnica de València video lectures (Polimedia)

launched its own MOOC platform available in https://www.upvx.es/ (see figure 1). The platform is powered by the edX[1] MOOC platform of MIT and Harvard University. The platform also includes a website for sharing video lectures https://media.upv.es (see figure 2).

The website provides a typical search engine allowing the students to search for learning objects (in this case videos) by indicating a set of keywords. For the moment, the searching process is just a comparison between keywords provided by users and the title of the video. The results consist of a set of videos that match the query of the student. This retrieval method does not take any semantic aspects into account in the search process. This approach can cause that many videos that are very appropriate but do not include in their title any of the searched keywords. In this way, useful videos will not be found and will not

[1]edX MOOC platform: https://www.edx.org/

be provided to students as a result. According to this, the work proposed in this paper deals with the improvement of the commented search engine introducing a new retrieval algorithm which can match the typed keywords with the main topics of videos and retrieve those that have a semantic similarity.

Usually, the discovery of the main topics in a set of documents reduces to finding the minimum number of clusters that separates the data in a way that the most similar documents (from the semantic point of view) are grouped in the same cluster. The literature in the area has been very prolific, and there are many clustering algorithms and methods [6], each tailored for specific types or amounts of data, domains, etc. However, if we need to determine specific keywords or a 'label/tag/class' to represent each cluster, the problem evolves to the supervised learning problem of 'classification'. Here, no matter the method selected, the classification accuracy must be evaluated either by comparing the results with a ground truth data-set (a data-set where each item is tagged with the information provided by direct observation, i.e. empirical evidence). However, ground truth data-sets must be manually created and/or curated by experts, which is hard and expensive work, unaffordable in a reasonable time for the case of large data-sets.

In this paper, we focus on the development of a classifier for educational videos from an unlabeled dataset. To do this, we have used third-party information to generate the embeddings and improve the dataset used for the classification process. We have used state-of-the-art embedding techniques that proved their effectiveness in other NLP tasks such as Question Answering [7], Natural Language Inference (MNLI), and others.

As the first step to refine the search engine of the media websites of the *Universitat Politècnica de València*, this work presents a semi-supervised method to cluster and classify the LO dataset of the university, by using open content resources from Wikipedia as labelled data to train the model [8]. According to our experiments presented in this paper, the proposed approach obtains a labelled dataset without the need of performing a manual data curation. This work is an improvement of a previous work presented in [9] where a more straightfor-

ward approach, using semi-supervised learning techniques, was tested to classify thousands of educational videos from the university's e-learning platform.

For clarity, we summarize the main contributions of this paper as follows: we propose a new method to correctly classify transcripts coming from educational videos available from UPV media; moreover, we implement a pipeline where we have improved the pre-processing of the information extracted from the videos; and finally, extensive experiments are carried out obtaining improved performance metrics comparing our proposal with different classification algorithms.

The paper is structured as follows: Section 2 presents the state-of-the-art on learning objects automatic clustering and classification. A transcript classification framework is proposed and described in Section 3 followed by Section 4 which provides the real-world dataset and the evaluation of the approach. Finally, Section 5 summarizes the works and concludes the article.

2. Related work

The continuous growth of e-learning platforms has encouraged the emergence of numerous educational recommendation systems (ERS). These systems provide help to students who are searching for resources to improve their knowledge in a specific domain [10]. This approach has been shown by the prolific series of European Conference on Technology Enhanced Learning[2]. The typical scenario is that of a student using a search engine in a website and typing some keywords to receive LOs recommendations. The accuracy of those recommendations can depend on many issues. Still, the critical point of any search engine or recommendation system is that the documents that it has to recommend are correctly modelled and classified as belonging to a specific topic. Therefore, topic modelling and LOs classification are crucial tasks.

In the best case, the algorithm can be trained to classify documents using a pre-tagged (ground truth) dataset, where each LO has a class (or a set)

[2]https://link.springer.com/conference/ectel

that represents the branch of knowledge to which it belongs (*supervised learning approach*). Thus, the community demand on large and diverse educational datasets was addressed by the *dataTEL Theme* Team of the Sustaining Technology Enhanced Learning Large-scale multidisciplinary Research European network of excellence (STELLAR[3]). Therefore, there were collected an initial set of datasets that have been used by the research community to develop such supervised learning models for ERS [11, 12]. However, the publicly available datasets, specially focused on Higher Education topics, are still scarce, small and mainly focused on learning resources in English. For clarity, we summarize the main contributions of this paper as follows: we propose a new method to correctly classify transcripts coming from educational videos available from UPV media; moreover, we implement a pipeline where we have improved the pre-processing of the information extracted from the videos; and finally, extensive experiments are carried out obtaining improved performance metrics comparing our proposal with different classification algorithms.

For this work, we have available a LO dataset of the Universitat Politècnica de València (Spain), which contains around 50.000 educational videos (UPV-Media dataset). The videos cover different subjects which are taught in the university and are usually presented by a lecturer in Spanish. Although they are not pre-tagged, more than 15.000 videos have a (Spanish) transcript as well as a set of associated keywords. Unfortunately, the keywords have a high level of noisy data and are not always linked to their knowledge domain, thus becoming difficult to model their topics and classify (tag) them as belonging to a specific knowledge domain, course, or subject. However, as explained in the previous section, the manual classification of LOs is a costly task that requires a lot of time of expert personnel in the different fields of knowledge. For our large LOs dataset, which is continuously growing, it is unfeasible.

Another option is to use a classification algorithm that does not require a ground truth dataset (*unsupervised learning approach*). The performance of

[3]https://cordis.europa.eu/project/id/231913

the systems based on this approach highly relies on the availability of large amounts of well-distributed data on which the model can be built [13]. Dealing with this limitation, in [14] authors address the problem to automatically annotate new LOs (cold-start problem) using a state-of-the-art automatic tag annotation method based on the Latent Dirichlet Allocation (LDA) probabilistic topic model, with an acceptable good performance with sparse and short textual content. Still, the unsupervised learning approach is computationally expensive and usually gets worse results due to the scarcity of learning resources in each cluster [15, 16].

Halfway through both approaches, *semi-supervised learning approaches* have been successfully applied to improve the learning accuracy of clustering methods (where no labelled data is available) by the usage of unlabeled data together with a small amount of labelled data available from other external sources from a related domain.

For instance, the work presented in [17] proposed a method of classifying Flickr tags as WordNet semantic categories by using Wikipedia articles for a first step classification and then mapping Flickr tags onto these tagged articles. In work proposed in [18], the authors identify the outcome and prerequisite concepts within a piece of educational content utilising a semi-supervised system that makes use of textbooks as external labelled data. The model is afterwards generalised to arbitrary web documents, thus without requiring expert intervention. In [19], a framework that uses human knowledge through labelling data from MOOCs and proposes a method for concept extraction based on machine learning (Conditional Random Fields) is presented. In this research, authors demonstrate that with only 10% of labelled data, the methods get an excellent performance. Finally, in [20], a new way to improve the accessibility of learning objects in educational websites by automatically enhancing the semantic metadata representations was proposed.

In a similar application domain, [21] tackled the same task (i.e., classification of unlabeled transcripts) with two key differences. One regards the fact that the corpus consists of transcripts from 12,032 video lectures from 200 courses

were collected from Coursera learning platform. The second difference regards the methodology, which used Word2Vec and Latent Dirichlet Allocation (LDA) to generate word embeddings and topic vectors. A comparative analysis cannot be performed because the transcripts are not publicly available as authors made available only the word embeddings and topic vectors.

To our knowledge, this work presents the first attempt to use an unsupervised classification method to automatically annotate a large dataset of Higher Education learning resources, getting promising results.

3. Proposed approach

The main objective of the proposed approach is to correctly classify transcripts coming from educational videos available from the UPV media LO dataset (UPV media). Current works extend the results from [9] which used a semi-supervised approach. The main issue regards the fact that the available dataset of transcripts is not labelled; that is, we do not have labels for transcripts or accurate keywords[4] and do not know the number of actual labels. Therefore, the usage of an unsupervised approach may lead to determining clusters of transcripts and thus, the topics in the dataset. However, still, the actual classification of a new transcript could not be possible.

The solution proposed in previous work implemented a semi-supervised approach which used Wikipedia articles as ground truth of correctly labelled data. This approach allowed the usage of available unlabeled data in a supervised context without the need of manually labelling a training dataset.

Taking into account that educational videos come from a controlled environment, we have taken into consideration three topics: *Biology*, *Engineering* and *Humanities*, which are the primary generic domains of the UPV courses. The initial solution used a semi-supervised approach based on a Support Vector Machine (SVM)[22] classification algorithm. The pre-training process of the model

[4] Actual keywords are manually typed as free text by the author of the LO, without following any standard or classification.

used Wikipedia articles from those different subcategories of biology, engineering and humanities. A custom build application [23] that uses the Wikipedia API [24] has been developed for retrieving the articles and building the training dataset.

The pre-processing of the Wikipedia articles, transcripts and keywords consisted of lower-casing the text and removing anything that did not represent a word. The next step consisted of computing the embeddings using a pre-trained model such that available texts were mapped to a 128-dimensional vector of numerical values that take into account the semantic value. Text embedding represents a necessary step since classification algorithms may process only numerical values. As embeddings require that the paragraphs keep their semantic value, there were performed no stemming or stop words removal.

The process of semi-supervised learning iteratively added to the training dataset the transcripts that were assigned the same label as the one determined for their corresponding keywords. This approach allowed that the final derived classifier has also been trained on transcripts, not only on Wikipedia articles. As the experimental results show, the accuracy, precision, recall and F1-score were reasonably good. The accuracy of the cross-validation trained model was 0.94, but the accuracy of the model on the unseen dataset was 0.87. We concluded that the prototype data analysis pipeline has a good quality, but the drop in validation accuracy requires a better data analysis pipeline. Current works present a more refined data analysis pipeline with improved validation results.

The limitations of the previous approach from [9] regard: (1) the methods used for computing the embeddings; (2) the algorithms used for classifying the transcripts and keywords; (3) the process for determining the transcripts which are being added to the training dataset; and (4) the validation methodology.

As several other state-of-the-art methods for determining the embeddings for a given paragraph were recently published, in this research we integrate them into an extensive benchmark for running experiments such that a detailed comparative analysis may be finally performed. Regarding the algorithm that performs the classification of both transcripts and keywords there are two im-

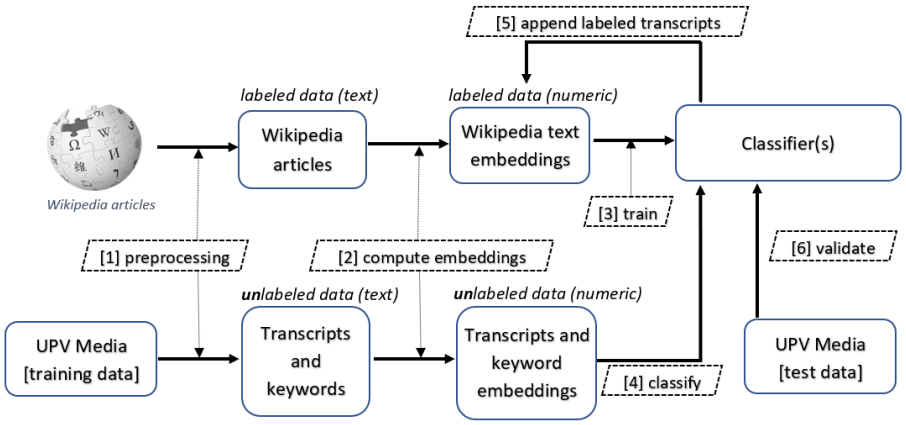

Figure 3: Overview of the data analysis pipeline

provements: firstly, the usage of other classification algorithms and secondly, the usage of co-training [25], that is the usage of distinct algorithms for training on transcripts and training on keywords. Finally, improvements in the validation methodology regard using new methods for choosing the test dataset and balancing the number of instances belonging to each class.

Taking into account the above outlined proposed solutions an updated data analysis pipeline has been designed and described in Figure 3 and presented in detail in Algorithm 1.

This new data analysis pipeline consists of the following key modules: (1) pre-processing; (2) embedding computation; (3) model training; (4-5) classify and append loop; and (6) model validation. The two main modules are the ones that compute the embeddings and the one that classifies and appends correctly labelled transcripts in the training dataset. These modules can be set up by specific word embedding algorithms, while classification may be performed by a wide range of algorithms with various hyperparameters tuning settings.

3.1. Data pre-processing

In this section, we introduce the necessary data pre-processing steps from the data analysis pipeline. This module includes a custom Wikipedia articles and keywords retrieval that builds a labelled dataset and a text processing of the UPV Media info that builds an unlabeled dataset of transcripts and keywords.

Retrieval of Wikipedia articles is performed by a custom-designed application [23] that scraps articles for the categories that are found under a specified topic. The Wikipedia articles are represented by *title*, *content*, *keywords* and *label* (i.e., the topic), covering thus all the discovered categories and thus obtaining a labeled training dataset. By proper settings of the Wikipedia scrapping application, a balanced dataset may be obtained such that each topic is roughly equally represented in the training dataset.

The second task of this module is to parse the transcripts and keywords available as metadata in the *.json* files from the UPV media dataset by (1) lower-casing all the words; (2) removing whatever is not considered a word; and (3) removing the outlier transcripts. From the data analysis perspective the transcripts that are being removed are considered outliers because they are instances that are very distant from the majority. The result consists of a dataset of transcripts along with their keywords.

3.2. Embedding computation

Training a classifier from the text (i.e., Wikipedia articles, keywords and transcripts) requires some form of numerical representation of the language. The critical issue is that classification algorithms need a representation of variables that they can understand and process. Hence, the typical operations required on the text are translation, categorization or questions-answering. From this perspective, since articles, keywords and video transcripts represent our input, we are in the area of text categorization.

Since language is in general ambiguous at the lexical and semantic level, it requires a proper representation of text such that the ML algorithms may handle it. The solution is to compute text embeddings by building a language model

such that we end up with numerical values (i.e., probabilities, frequencies, etc.) instead of words, paragraphs or entire documents.

Our proposed approach for embedding computation relies on state-of-the-art distributional semantics which represents text as dense vectors that may be successfully included as features of training data for ML algorithms. The main advantage of this approach regards the fact that the semantic similarity between two texts may be computed as cosine distance between their corresponding embeddings.

Therefore, our solution for embedding computation reduces to custom integration of state-of-the-art language models within the data analysis pipeline such that the final validation metrics have the best possible values for our real-world datasets.

3.3. Train, classify and append loop

Algorithm 1 presents in detail the data analysis pipeline that is executed after the embeddings computation is performed (i.e., see the overview of the data analysis pipeline presented in Figure 3 and the embedding computation module described in subsection 3.2). The *Wiki-Dataset* and *Transcripts-Dataset* word embeddings represent the *input* for Algorithm 1. From the ML perspetive, the *Wiki-Dataset* represent the ground-truth that is initially used to create *Model-X* by training $\boldsymbol{C}$ classifier (i.e., SVM, Random Forest or XGBoost). More exactly, the *Wiki-Dataset* is split into *Wiki-Train*, *Wiki-Test* and *Wiki-Validate* datasets (i.e., 70%, 15% and 15%, respectively) such that only the *Wiki-Train* dataset is used by the employed classifier within the *while* loop for obtaining the initial *Model-X* model.

Thereafter, within the *while* loop there is a *for* loop which uses *Model-X* to predict the label of each *(transcript, keyword)* pair, and if they have the same label then the *transcript* is appended to the *Wiki-Train* dataset. The *while* loop is controlled by the *threshold* parameter, which represents the number of transcripts that are appended at each iteration. Once the number of valid *(transcript, keyword)* pairs that are being appended at last iteration is small

enough (i.e., less than the *threshold*) we conclude that semi-supervised training loop needs to end.

Monitoring of the accuracy of *Model-X* is performed within the *while* loop by cross-validating the classifier on *Wiki-Train* dataset after each training. More, the current model is also validated against unseen data from the *Wiki-Test* dataset. This approach controls the unsupervised training loop and prevents the over-fitting of *Model-X*. At the end of the *while* loop the *Wiki-Train* dataset will also contain the labeled transcripts and *Model-X* is trained on this *Wiki-Train* augmented dataset.

The next step in the data analysis pipeline uses *Model-X* to predict labels for the remaining transcripts, that is transcripts that were not appended to the *Wiki-Train* dataset.

At the end of the *while* loop we end up with the *Transcripts-Labeled* dataset, which consists of all valid transcripts labeled by *Model-X*. The current approach does not use in any way the transcripts and keywords have not the same label. There may be various reasons for which a transcript is not labelled the same as its keywords, but the main issue that may explain this situation is the poor quality of the keywords that were associated with the transcript.

Finally, with all the labelled transcripts, we build *Model-Y*, a model that is trained on the newly generated *Transcripts-Labeled* dataset. This model is validated against the *Wiki-Validate* dataset that represents unseen data during training and testing of *Model-X*. This final validation provides the most conclusive intuition regarding the quality of the designed data analysis pipeline. The situation in which the validation of *Model-Y* on unseen *Wiki-Validate* dataset obtains good scores is a clear indication that the previously inferred *Model-X* also has high quality. On the contrary, good results in validation of *Model-X* on *Wiki-Test* along with poor results of *Model-Y* on *Wiki-Validate* represents a clear indication that semi-supervised training has been particularised by the appended transcripts and does not generalize backwards to Wikipedia articles.

Algorithm 1 Data analysis pipeline

Require: Wiki-Dataset = Wiki articles and keywords embeddings as labeled dataset

Require: Transcripts-Dataset = transcripts and keywords embeddings as unlabeled dataset and keywords

$Wiki - Train \leftarrow 70\%$ of $Wiki - Dataset$

$Wiki - Test \leftarrow 15\%$ of $Wiki - Dataset$

$Wiki - Validate \leftarrow 15\%$ of $Wiki - Dataset$

while (valid transcripts greater than *threshold*) **do**

  Model-X = Train $\boldsymbol{C}$ classifier on Wiki-Train

  # Cross-Validate Model-X on Wiki-Train

  # Validate Model-X on Wiki-Test

  {This is for continuous monitoring of the training process.}

  for all (**transcripts & keywords** pairs in Transcripts-Dataset) **do**

    #Predict label of *transcript* and *keyword* using Model-X

    if (*transcript* and *keyword* have same label) **then**

      #Append *transcript* to Wiki-Train

    end if

  end for

end while

{ After while loop:

- Wiki-Train contains the labeled appended transcripts.

- Model-X is build final Wiki-Train, the one that contains also the labeled transcripts. }

Transcripts-Labeled = labeled transcripts during while loop by final Model-X

Model-Y = Train classifier on Transcripts-Labeled

#Validate **Model-Y** on Wiki-Validate

3.4. Data analysis pipeline validation

The overall data analysis pipeline has a multi-step validation methodology that will be further presented in detail. As a general approach, there are three points at which validation is performed, and these may also be found within Algorithm 1.

1. *Model-X training and monitoring* is performed after each append of transcripts to the *Wiki-Train* dataset. The iterative process of appending labelled transcripts to the *Wiki-Train* dataset is monitored by cross-validating the current challenger model. Thus, after each iteration there are logged the number of valid transcripts (i.e., transcripts that had the same label as their keywords) that will be added to the *Wiki-Train* dataset along with the remaining number of transcripts. Further, upon training on the newly obtained dataset, the *Model-X* is retrained, and accuracy metrics (i.e., accuracy, precision, recall and f1-score) are determined. This approach makes possible proper debugging of the training process in terms of stopping criteria, accuracy metrics monitoring, and finally discovering the best available *Model-X* that will be further used. From the data analysis perspective, we hypothesise that the initial model is trained exclusively on the *Wiki-Train* dataset to have the highest accuracy among all other obtained models after appending labelled transcripts. We expect that each append of transcripts to slightly decrease the accuracy of *Model-X*.

The motivation for following this approach regards the need to observe how much the cross-validation accuracy changes after each iteration, or more precisely, to see how sensible is the model to the transcripts that are being added. This approach is not used in any way for validating the *Model-X*, but to check its stability. Monitoring the accuracy levels regards the internal logging and debugging of the training process as situations of high accuracy drop need to identified and further investigated in detail.

The critical issue is to determine and avoid a sudden drop in validation metrics and eventually investigate the reasons for such a reduction. One reason which may decrease the accuracy may be the low quality of keywords. This situation gives rise to many missclassifications that are transcripts which were

correctly classified but were not appended to the *Wiki-Train* dataset due to the incorrect classification of their corresponding keywords. This scenario is quite probable since keywords were manually set by content creators and therefore may not always actually represent the information from the transcript.

2. *Model-X validation* that is performed *Wiki-Test* dataset in the semi-supervised training loop. Once the *Model-X* is determined within the *while* loop it is tested against over-fitting on the unseen *Wiki-Test* dataset. The validation metrics determined at this step may be considered final benchmark values and represent the true values that we expect when *Model-X* is deployed in production. The limitation of this approach consists of the fact that final *Model-X* takes into consideration only valid instances from the transcripts dataset, that is transcripts whose label is the same as the one determined for its keywords. Still, taking into account that we started with a completely unlabeled transcripts dataset and considered as ground-truth the labelled Wikipedia dataset performing a the *Model-X* validation on the *Wiki-Test* dataset provides an insight regarding the over-fit that has been done within the *while* loop and which may not be determined by cross-validation.

3. *Model-Y validation* that is performed on *Wiki-Validate*. *Model-Y* is the model that is trained on valid labelled transcripts obtained within the *while* loop of the data analysis pipeline. We perform the validation of *Model-Y* against the *Wiki-Validate* dataset that has not been used in any way in the data analysis pipeline so far. In this way, we evaluate the quality of the predicted transcripts labels against the unseen ground-truth dataset represented by *Wiki-Validate*. This final validation step represents a clear indication regarding the quality of the final *Model-X* by comparing the obtained validation metrics with the ones obtained from the previous *Model-X validation* step. The interpretation is straight forward. Similar values in metrics indicate that the data analysis pipeline has been well designed and produced a final reliable model. On the other hand, a decrease on the metrics in *Model-Y* validation as compared with *Model-X* validation represents a clear indication that we are in an over-fit situation and training-validation from the *while* loop is not reliable. This scenario

may be due to many factors which need further detailed investigation.

4. Experimental Results and Evaluation

4.1. Wikipedia articles and transcripts datasets

The Wikipedia scrapping tool build a dataset of 3747 articles: 1219 with topic *Biology*, 1060 for *Engineering* and the rest 1468 from *Humanities*. Under the topic of *Biology* there are nine categories: *Biologia*, *Anatomia*, *Bioinformatica*, *Biologia celular*, *Bioquimica*, *Biotecnologia*, *Botanica*, *Microbiologia* and *Genetica*. Within **Engineering** Wikipedia provides eight categories: *Ingenieria*, *Materiales en ingenieria*, *Bases de datos*, *Computacion distribuida*, *Computación grafica*, *Geomatica*, *Ingenieria de software*,*Seguridad informatica* and within **Humanities** the categories are *Arte*, *Tecnicas de arte*, *Antropologia*, *Simbolos*, *Ciencias Historicas*, *Ciencias sociales*, *Economia*, *Sociologia*, *Comunicacion*, *Terminos juridicos*, *Justicia*, *Derecho*, and *Principios del derecho*.

The wikipedia dataset contains 4 features. The page id of the wikipedia article, the title of the article, the content of the article, the keywords of the article and the class of the article. The wikilinks from a page were considered as keywords and this choice was justified by the fact that in the wikipedia guidelines is specified that wikilinks should be created when they provide relevant connections to the subject. The content of the article was extracted using the Wikipedia python library without any preprocessing done on it. Even though the wikipedia keywords were not used in the pipeline presented above, they were used in co-training as a different view of the wikipedia dataset. In figure 4 we present a histogram with the number of articles in each class.

4.1.1. Transcripts dataset description

The total number of available educational videos in UPV Media is about 50.000. The videos cover various subjects that are taught at UPV and are usually presented by a lecturer in Spanish. Along with the educational video itself, the metadata may also contain a title, several slides, keywords, the duration

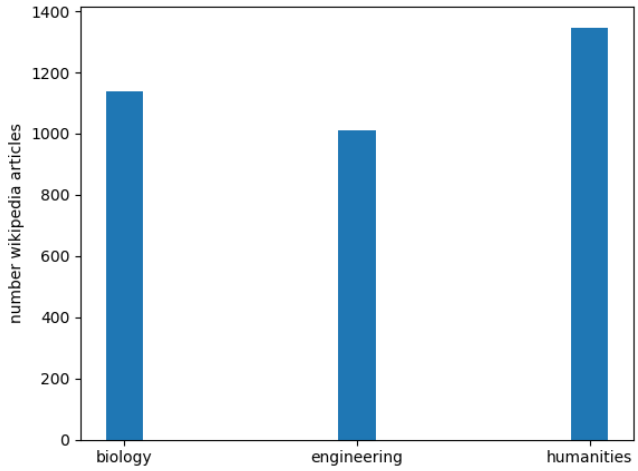

Figure 4: Number of Wikipedia articles per class

and possibly the transcript. The slides are screen-shots from the video which are chosen by the presenter as a summary for the video. The manual analysis of several videos revealed that keywords are not always representative for the main topics or domain of the video, so their usage in the data analysis may be misleading. Therefore, the learning objects consist of educational videos, with several slides (i.e., screen-shots from the video) which are chosen by the presenter as a summary of the video, and sometimes, the transcript of what the lecturer is saying. A sample of the metadata is presented in the example below:

```
{'_id': '00054a38-5a32-4db2-ae9c-85c296015c3b',
 'hidden': False,
 'title': 'Programa Mathematica gratis y online',
 'source': {'type': 'polimedia',
            'videos': [{'mimetype': 'video/mp4',
                        'width': 640, 'height': 480
                        'src': 'politube.mp4',
                         }],
            ...},
 'slides': [
```

```
    {'mimetype':'image/jpg','url':'frame.0.jpg','time':'0'},
    {'mimetype':'image/jpg','url':'frame.48.jpg','time':'48'},
    ...
    {'mimetype':'image/jpg','url':'frame.432.jpg','time':'432'}
  ],
  "metadata": {
            "keywords": [
                "croma"
            ]
        },
  'duration': 564.523537,
  'transcription': '...very long text... (or empty)'
}
```

As we can see, there is no parameter which specifies from what field the videos are (e.g. Biology, Engineering, Humanities, etc.). The only parameter which could be useful in finding the class to which the video belongs to is the transcription, but this is not available for all the videos (just 15.387 videos have a transcription and keywords attached).

Even when transcriptions are available, there are problems regarding their quality. Some transcriptions seem to have no punctuation since most of them were auto-generated by a speech recognition engine. This practical scenario makes the task of separating a transcription in sentences a challenging one.

For this work, we processed the original dataset and created a new dataset with each video's transcription and the keywords. Figure 5 shows the IQR (Interquartile range) analysis for the transcripts and Wikipedia articles. Based on this analysis, transcripts and Wikipedia articles with less than ten words were removed since may contain too few information such that it may be reliably used for training or testing. In the same way, transcripts and Wikipedia articles with more than 2.800 words were also considered outliers by the IQR analysis and therefore were removed. An interesting observation is that IQR thresholds (Q1, Median and Q3) had similar values for both transcripts and Wikipedia articles.

We decided to have 3 classes/domains that cover the main topics of the courses of UPV. The motivation for this choice was the fact that it is hard to

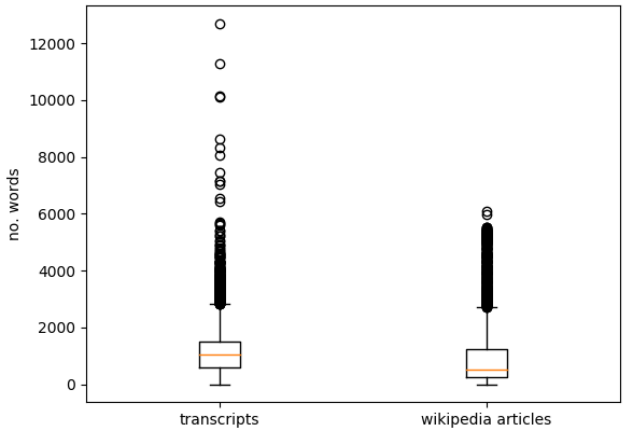

Figure 5: Interquartile range analysis on the size of transcripts and Wikipedia articles

determine an estimation of the number of subdomains in the UPV dataset and to create a dataset of wikipedia articles for each subdomain. The deeper you go in the subdomain dependency tree, the lesser the number of articles becomes. From each domain we chose the most popular subdomains with the most pages to have enough information to classify a transcript in the specific domain, and we used a scrapping tool to extract all the articles from that subdomain. Note that it is easier to differentiate between distinct domains than between subdomains. The topics from Wikipedia articles represent the labels for the loop in data analysis pipeline (Algorithm 1). The wikipedia articles dataset is a labeled dataset, where labels are represented by the topics.

Regarding the keywords, which come with the transcripts in json, they were entered by the authors. In the end of the pipeline, some transcripts may not be added to the labelled dataset because the quality of the keywords may be poor (some professors may use their names as keywords) or misclassifications.

4.2. Algorithms for text embedding and classification

The pre-built text embedding methods were feed-forward Neural-Net Language Models[26] with pre-built Out Of Vocabulary terms (OOV) from *ten-*

sorflow hubs feed-forward Neural-Net Language Models implemented in [27], universal sentence encoder [28] implemented in [29] and BERT [30]. The training has been performed by three classification algorithms: Random forest (RF), Extreme Gradient Boosting (XGBoost) and Support Vector Machines (SVM).

The Universal Sentence Encoder uses a Convolutional Neural Net (CNN) architecture and covers 16 languages (Spanish as well). Input sentences are truncated to 256 tokens for the CNN model and 100 tokens for the transformer. The CNN encoder uses 2 CNN layers with filter width of [1, 2, 3, 5] and a filter size of 256. Average pooling is used to turn the token-level embeddings into a fixed length representation of size 512. The model was trained on QA pairs from online forums and QA websites (Reddit, StackOverflow, Yahoo Answers), mined translation pairs and the Stanford Natural Language Inference dataset.

The BERT model uses the transformer architecture, with 12 hidden layers, a hidden size of 768 and 12 attention heads. The model was pre-trained on the multilingual wikipedia and it makes the distinction between lower case, upper case and accent markers. The Universal Sentence Encoder is pre-trained on a dataset that emphasis more the semantic similarity while the BERT embeddings are pre-trained on a multilingual wikipedia dataset. The results that used the Universal Sentence Encoder embeddings are similar to the ones which used the BERT ones, which is a huge gain taking into consideration that the computation time is much faster using the Universal Sentence Encoder.

Regarding the classifiers, Random Forest uses the default hyper-parametrization from sklearn library but with the exception of the number of estimators which is set on 50. The XGBoost uses the default hyperparametrization from the XGBoost python library, while the SVM uses auto for gamma, the one vs one decision function, the RBF kernel and the regularization parameter equal to 5.

4.3. Numerical results and discussion

Table 1 shows the results of each iteration of the training process, with the number of valid transcripts added to the dataset at each iteration (labelled in column head as *valid*) along with the number of available transcripts. The

second column presents the cross-validation (CV) score along with the average of F1-score (F1-avg) of the validation of the trained model against unseen data in *Wiki-Test* dataset. Last columns present the detailed classification accuracy metrics by class obtained when validating the model on the *Wiki-Test* dataset. The slight decrease of CV values in the second column is due to the continuous adding of valid transcripts in the training dataset. The reasonably constant values from F1-avg show that the rebuilt model still generalizes well on *Wiki-Test* unseen data after each append of transcripts.

Tables 2, 3 and 4 present the cross-validation results of the trained models from BERT, NNLM and USE embeddings along with the validation results on the unseen *Wiki-validate* dataset.

The main conclusion is that the best result has been obtained using USE embeddings with XGBoost classifier. The 91% average F1-score strengthens the obtained cross-validation accuracy of 94% of the best-trained model performed on the last validation step on the *Wiki-Train* dataset. Regarding the classifiers, the decreasing order of accuracy is XGBoost, SVM and Random Forest. In terms of embedding computation, the best results were obtained by USE, while the most unsatisfactory results were obtained by NNLM. The advantage of USE and BERT over NNLM regard the fact that the former one captures only a general meaning of the text, while the first ones have a deep understanding of the context.

Experimental results were performed by co-training, that is using two distinct classifiers in the *while* loop of the semi-supervised learning process, one for transcripts and one for keywords. The intuition behind this alternative approach is that each classifier may separately be trained and thus capture a distinct aspect of the underlying data. As numeric results obtained by co-training are similar to the ones obtained by using single trainers, we conclude that experimental results are reliable and may be reliably used in practice.

For running the data analysis pipeline, some technical issues were met. One of them regards the fact that not all pre-trained models for embeddings worked on the same operating system. For example, BERT and NNLM are available on

#iter (valid/available)	CV/F1-avg	Class	Precision	Recall	F1-score
1(7708 / 14920)	0.88/0.90	Biology	0.93	0.91	0.92
		Engineering	0.87	0.88	0.87
		Humanities	0.91	0.92	0.91
2(3610 / 7212)	0.86/0.90	Biology	0.94	0.92	0.93
		Engineering	0.86	0.84	0.85
		Humanities	0.89	0.92	0.90
3(1582 / 3602)	0.85/0.89	Biology	0.93	0.88	0.91
		Engineering	0.85	0.86	0.86
		Humanities	0.89	0.92	0.90
4(768 / 2020)	0.85/0.90	Biology	0.93	0.88	0.91
		Engineering	0.86	0.88	0.87
		Humanities	0.89	0.92	0.90
5(461 / 1252)	0.85/0.88	Biology	0.92	0.85	0.89
		Engineering	0.85	0.85	0.85
		Humanities	0.86	0.92	0.89
6(352 / 791)	0.85/0.89	Biology	0.95	0.89	0.92
		Engineering	0.86	0.86	0.86
		Humanities	0.86	0.92	0.89
7(111 / 439)	0.85/0.87	Biology	0.90	0.89	0.89
		Engineering	0.85	0.79	0.82
		Humanities	0.86	0.92	0.89
8(70 / 328)	0.85/0.88	Biology	0.93	0.88	0.90
		Engineering	0.85	0.85	0.85
		Humanities	0.87	0.92	0.89
9(47 / 258)	0.85/0.88	Biology	0.93	0.87	0.90
		Engineering	0.83	0.86	0.85
		Humanities	0.88	0.91	0.89

Table 1: Cross-validation scores after each append of correctly classified transcripts. The word embeddings are computed by BERT and training is done by Random Forest classifier

Classifier	#iter	CV/F1-avg	Remaining transcripts	Class	P	R	F1
RF	9	0.86/0.77	211	Biology	0.97	0.58	0.73
				Engineering	0.65	0.83	0.73
				Humanities	0.76	0.91	0.83
XGB	7	0.90/0.86	2992	Biology	0.91	0.84	0.88
				Engineering	0.81	0.87	0.84
				Humanities	0.85	0.88	0.87
SVM	5	**0.94/0.9**	2726	Biology	0.93	0.93	0.93
				Engineering	0.85	0.92	0.89
				Humanities	0.92	0.87	0.89

Table 2: Validation results by using BERT embeddings on RF, XGB and SVM classifiers

Classifier	#iter	CV/F1-avg	Remaining transcripts	Class	P	R	F1
RF	11	0.88/0.73	267	Biology	0.98	0.55	0.70
				Engineering	0.78	0.66	0.71
				Humanities	0.61	0.95	0.75
XGB	8	0.91/0.77	3213	Biology	0.96	0.60	0.74
				Engineering	0.64	0.86	0.73
				Humanities	0.78	0.87	0.82
SVM	8	**0.92/0.86**	1984	Biology	0.91	0.85	0.88
				Engineering	0.77	0.89	0.83
				Humanities	0.89	0.85	0.87

Table 3: Validation results by using NNLM embeddings on RF, XGB and SVM classifiers

Classifier	#iter	CV/F1-avg	Remaining transcripts	Class	P	R	F1
RF	10	0.90/0.82	334	Biology	0.98	0.69	0.81
				Engineering	0.74	0.87	0.80
				Humanities	0.78	0.91	0.84
XGB	6	**0.94/0.91**	2282	Biology	0.93	0.95	0.94
				Engineering	0.88	0.88	0.88
				Humanities	0.92	0.90	0.91
SVM	6	0.88/0.58	2131	Biology	1.00	0.01	0.02
				Engineering	0.87	0.80	0.83
				Humanities	0.48	0.98	0.65

Table 4: Validation results by using USE embeddings on RF, XGB and SVM classifiers

Windows, but USE is available only on Linux (i.e., on TensorFlow-hub). Using the models on Windows is possible through Linux Subsystem For Windows, but currently, there is no support for GPU. This issue can be solved by using language models (i.e., BERT) available on Windows as a REST Service. The entire computation took approximately two days. The pre-trained language models were not the bottlenecks since NNLM and USE were fast, and BERT could be run by using the REST Service. Most computation time was due to the training algorithms and a large number of iterations. The TensorFlow-text library is available just on Linux. In this case, the new models which are trained on TensorFlow 2 and benefit from this library can't be used on Windows without the Linux Subsystem for Windows.

5. Conclusions and future work

In this work, we address the challenge of classifying thousands of educational videos from a university's e-learning platform (*Universitat Politècnica de*

València). These videos need to be classified to improve the search and recommendation mechanisms and thus provide a better service to students, with learning objects that are better adapted to their needs and preferences.

In this paper, a previous work where semi-supervised learning is used to classify the videos has been improved. This approach was selected to deal with the problem of not having a tagged and validated dataset. Instead, we have only the metadata of the videos, where the most relevant one is the information used in this work, which is the automatic transcription of the video lessons.

With this information and using Wikipedia as a knowledge base to train the models, an improved pipeline has been built where the data from the videos is pre-processed. The pipeline has been used to improve the trained model from Wikipedia in three categories. In the pipeline presented in this paper, the validation methodology has also been developed, as well as the comparison with different classification algorithms, obtaining; as a result, improved performance metrics.

In conclusion, a classifier for educational videos has been generated from an unlabeled dataset and from which we do not have ground truth. The work presented here addresses these problems by using third-party information to create the embeddings and improve the dataset used for classification. Therefore, this proposal allows obtaining a labelled dataset without the need of performing a manual data curation.

As future work, we intend to validate the keywords included in the video metadata (which have been proven to be often not relevant) using the techniques proposed in this work. Besides, to further improve the validation of the models, we want to create a dataset manually labelled by native Spanish speakers to be used in future work.

Further improvements may take into consideration other input datasets for pre-training the language model such as scientific articles, open book corpus or a domain ontology.

Acknowledgements

This work was partly supported by the Generalitat Valenciana (PROMETEO/2018/002), by the Spanish Government (RTI2018-095390-B-C31), and by the Erasmus+ scholarship RO CRAIOVA01 awarded to Alexandru Stefan Stoica.

References

[1] C. H. Papadimitriou, P. Raghavan, H. Tamaki, S. Vempala, Latent semantic indexing: A probabilistic analysis, Journal of Computer and System Sciences 61 (2) (2000) 217–235.

[2] M. Steyvers, T. Griffiths, Probabilistic topic models, Handbook of latent semantic analysis 427 (7) (2007) 424–440.

[3] H. M. Wallach, Topic modeling: beyond bag-of-words, in: Proceedings of the 23rd international conference on Machine learning, ACM, 2006, pp. 977–984.

[4] S. Downes, Learning objects: resources for distance education worldwide, The International Review of Research in Open and Distributed Learning 2 (1) (2001).

[5] B. Tucker, The flipped classroom, Education next 12 (1) (2012) 82–83.

[6] P. Berkhin, A survey of clustering data mining techniques, in: Grouping multidimensional data, Springer, 2006, pp. 25–71.

[7] G.-S. Pîrtoacă, T. Rebedea, S. Ruseti, Answering questions by learning to rank–learning to rank by answering questions, arXiv preprint arXiv:1909.00596 (2019).

[8] S. Overell, B. Sigurbjörnsson, R. Van Zwol, Classifying tags using open content resources, in: Proceedings of the Second ACM International Conference on Web Search and Data Mining, ACM, 2009, pp. 64–73.

[9] A. S. Stoica, S. Heras, J. Palanca, V. Julian, M. C. Mihaescu, A semi-supervised method to classify educational videos, in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2019, pp. 218–228.

[10] H. Drachsler, K. Verbert, O. C. Santos, N. Manouselis, Panorama of recommender systems to support learning, in: Recommender systems handbook, Springer, 2015, pp. 421–451.

[11] K. Verbert, N. Manouselis, H. Drachsler, E. Duval, Dataset-driven research to support learning and knowledge analytics, Journal of Educational Technology & Society 15 (3) (2012) 133–148.

[12] S. Fazeli, B. Loni, H. Drachsler, P. Sloep, Which recommender system can best fit social learning platforms?, in: European Conference on Technology Enhanced Learning, Springer, 2014, pp. 84–97.

[13] R. Krestel, P. Fankhauser, Language models and topic models for personalizing tag recommendation, in: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol. 1, IEEE, 2010, pp. 82–89.

[14] E. Diaz-Aviles, M. Fisichella, R. Kawase, W. Nejdl, A. Stewart, Unsupervised auto-tagging for learning object enrichment, in: European Conference on Technology Enhanced Learning, Springer, 2011, pp. 83–96.

[15] B. Batouche, A. Brun, A. Boyer, Unsupervised machine learning based on recommendation of pedagogical resources, in: European Conference on Technology Enhanced Learning, Springer, 2014, pp. 548–549.

[16] B. Batouche, A. Brun, A. Boyer, Clustering based recommendation of pedagogical resources, Challenges for Research into Open & Distance Learning (2014).

[17] X. J. Zhu, Semi-supervised learning literature survey, Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2005).

[18] I. Labutov, Y. Huang, P. Brusilovsky, D. He, Semi-supervised techniques for mining learning outcomes and prerequisites, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 907–915.

[19] Z. Jiang, Y. Zhang, X. Li, Moocon: a framework for semi-supervised concept extraction from mooc content, in: International Conference on Database Systems for Advanced Applications, Springer, 2017, pp. 303–315.

[20] K. Niemann, Increasing the accessibility of learning objects by automatic tagging, in: Proceedings of the Fifth International Conference on Learning Analytics And Knowledge, 2015, pp. 414–415.

[21] Z. Kastrati, A. Kurti, A. S. Imran, Wet: Word embedding-topic distribution vectors for mooc video lectures dataset, Data in brief 28 (2020) 105090.

[22] L. Wang, Support vector machines: theory and applications, Vol. 177, Springer Science & Business Media, 2005.

[23] A. Stoica, Wikipedia page extractor, https://github.com/Arkin1/Valencia-Educ-Video (2019).

[24] J. Goldsmith, Wikipedia api for python, https://pypi.org/project/wikipedia/ (2019).

[25] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the eleventh annual conference on Computational learning theory, 1998, pp. 92–100.

[26] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, Journal of machine learning research 3 (Feb) (2003) 1137–1155.

[27] Google, Token based text embedding trained on Spanish Google News 50B corpus, https://tfhub.dev/google/nnlm-es-dim128-with-normalization/2, accessed: 2020-01-31 (2019).

[28] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y.-H. Sung, et al., Multilingual universal sentence encoder for semantic retrieval, arXiv preprint arXiv:1907.04307 (2019).

[29] Google, Saved Model: universal-sentence-encoder-multilingual-qa, https://tfhub.dev/google/universal-sentence-encoder-multilingual-qa/3, accessed: 2020-01-31 (2019).

[30] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).