computations

MDPI

Article

Comparison and Evaluation of Different Methods for the Feature Extraction from Educational Contents

Jose Aguilar [1,2,*,†], Camilo Salazar [2,†], Henry Velasco [3,†], Julian Monsalve-Pulido [2,†] and Edwin Montoya [2,†]

1 Escuela de Sistemas, Facultad de Ingeniería, Universidad de los Andes, Mérida 5101, Venezuela
2 GIDITIC, Universidad EAFIT, Carrera 49 No. 7 Sur 50, Medellin 050001, Colombia; jhenaos@eafit.edu.co (C.S.); jmsistemas@gmail.com (J.M.-P.); emontoya@eafit.edu.co (E.M.)
3 LANTIA SAS, Medellin 050001, Colombia; hgvelascov@eafit.edu.co
* Correspondence: aguilar@ula.ve; Tel.: +58-4265742164
† These authors contributed equally to this work.

Received: 17 February 2020; Accepted: 23 March 2020; Published: 15 April 2020

Abstract: This paper analyses the capabilities of different techniques to build a semantic representation of educational digital resources. Educational digital resources are modeled using the Learning Object Metadata (LOM) standard, and these semantic representations can be obtained from different LOM fields, like the title, description, among others, in order to extract the features/characteristics from the digital resources. The feature extraction methods used in this paper are the Best Matching 25 (BM25), the Latent Semantic Analysis (LSA), Doc2Vec, and the Latent Dirichlet allocation (LDA). The utilization of the features/descriptors generated by them are tested in three types of educational digital resources (scientific publications, learning objects, patents), a paraphrase corpus and two use cases: in an information retrieval context and in an educational recommendation system. For this analysis are used unsupervised metrics to determine the feature quality proposed by each one, which are two similarity functions and the entropy. In addition, the paper presents tests of the techniques for the classification of paraphrases. The experiments show that according to the type of content and metric, the performance of the feature extraction methods is very different; in some cases are better than the others, and in other cases is the inverse.

Keywords: feature extraction; content analysis; educational contents; semantic representation; information retrieval; recommendation system

1. Introduction

The growth of the internet in recent years and the emergence of multiple sources of information has led to the construction of new models for searching, retrieving and classifying information, through the application of specific techniques according to the domain of application. In the educational domain, the information available in digital media has significantly increased, due to the extended use of virtual learning environments (VLEs) in learning processes. Currently, students only need an internet connection and a device to be able to enter at any time and in any place into an academic platform, with digital content methodologically adapted to the teaching-learning processes. Academic digital resources have evolved through time as a consequence of three main factors: the need of updating academic subjects through time, the diversity of channels that students use to consume academic content, and the technical and pedagogical quality required to be included in VLEs. Digital resources are available in diverse repositories, such that extraction, classification, recommendation mechanisms are required to be used by a VLE [1]. For the location, development, classification, combination,

installation and maintenance of digital resources in the VLEs, it is necessary specialized tasks that use metadata.

Particularly, a challenging task is to build semantic representations of educational digital resources. This construction is required by several educational systems, for example, learning object recommendations systems, or educational information retrieval systems, both very important in the context of VLEs. These semantic representations can be defined by the features/characteristics of the digital resources [2,3]. Particularly, it is possible to use some of the Learning Object Metadata (LOM) fields [4], like the title, descriptions, among others, like short texts to be used for the construction of the semantic representation.

This article analyzes the semantic representation of diverse approaches for feature extraction in the educational domain: a statistical technique (Best Matching 25 (BM25)), a method of vectorization of documents (Latent Semantic Analysis (LSA)), a neural network method (Doc2Vec), and a probabilistic method (Latent Dirichlet allocation (LDA)). They are tested on different types of educational digital resources (scientific publications, learning objects, patents), on a paraphrase corpus, and in two contexts of utilization of the features: an information retrieval system and an educational recommendation system. During the analysis of the datasets, unsupervised metrics are used to determine the feature quality proposed by each one. In the case of the paraphrase corpus, performance metrics of the classification problems are used.

The main contributions of this paper are: (a) the analysis of different methods for the feature extraction from educational contents; (b) the study of different types of educational digital resources; (c) the utilization of unsupervised metrics to determine the feature quality proposed by each one, (d) the analysis of two use cases: information retrieval and educational recommendation systems, and finally, (e) the study of the different techniques in the context of a classification problem: the classification of paraphrases. Particularly, the selection of the datasets, methods, metrics and use cases are due to:

- In the case of datasets, we have selected three typical types of educational digital resources that can be modeled using the LOM standards, and of which there are repositories from which they can be extracted to be used in a VLE (scientific publications, learning objects, patents).
- In the case of extraction methods, we have selected methods with different theoretical basis (deep learning, frequency, probabilities and vector analysis), in order to test the capabilities of each theory.
- The performance metrics used allow the self-evaluation of the quality of the results proposed for each method, without requiring a comparison with a reference group (like it is the case in a supervised context).
- Finally, the use cases studied are two cases very useful in the context of a VLE: the recommendation systems to bring educational digital resources, and the information retrieval systems to search personalized information.

The document is organized as follows: Section 2 will present related works to this research, with a comparison with our proposal. Section 3 briefly describes the strategies used in this paper for feature extraction. Section 4 presents three evaluation processes: the first one uses unsupervised metrics like similarity functions and entropy to establish the quality of each feature extraction method; the second one analyses a classification problem; and the third one analyses two use cases. Finally, some conclusions and future works are presented.

2. Related Works

2.1. Literature Review

Fano, Karlgren and Nivre [5] evaluate the performance of three different types of semantic vectors or word embeddings (random indexing, GloVe, and ELMo), for the identification of persons with eating disorders from the writings they published on a discussion forum. This paper used the Early

Risk Prediction on the Internet (eRISK) dataset, which was used in the Conference and Labs of the Evaluation Forum (CLEF) 2019. They did not observe an advantage with the utilization of ELMo, compared to the commonly used, like GloVe or the random indexing approach. Singh et al. [6] propose a vectorization approach based on word targets, to identify unifiable news articles. They define a framework for identifying news related to trending topics/hashtags. Then, they carry out a multi-document summarization of unifiable news based on the trending topics. Previously, they put the corpus of news related to each trending topic through a text clustering, in order to obtain smaller unifiable groups. They analyse the effectiveness of various text vectorization methods, such as the bag of word representations with tf-idf scores, word embedding, and document embedding, using the k-means algorithm, the Document Understanding Conferences (DUC) 2004 benchmark dataset, and the purity metric.

Peng et al. [7] obtained a document-topic vector representations by combining LDA and Topic2Vec, and then, they perform document representations based on the topic vectors and the document vectors obtained through a trained Doc2Vec. They use their approach for document classification tasks. In [8], they propose the Topic2Vec approach that can learn topic representations in the same semantic vector space of words. The experimental results show that Topic2Vec achieves interesting and meaningful results. Ritu et al. [9] discuss the performance of word2vec in Tensorflow, in Gensim (Python library for topic modelling, document indexing and similarity retrieval) and FastText model, on a Bangla dataset containing 5,21,391 words, and they evaluate their performance in terms of accuracy and efficiency. They determine that FastText- Skip Gram model produces the best results. The authors of [10] analyse the quality of biterm topic modeling (BTM) and the word embedding approaches in the Gensim library, in a set of suggestions about disaster risk reduction strategies, provided by residents in disaster-prone areas of the Philippines. A word intrusion test was conducted, and BTM gives a strong cohesion of the words with their topics. For word embedding, the word2vec results have a high cosine similarity, which implies strong relatedness of each word.

Kadhim presents a comparative study of two feature engineering techniques, BM25 and Term Frequency-Inverse Document Frequency(TF-IDF), to weight the terms on Twitter [11]. Its experiments show that TF-IDF has the best performance, according to the value of F1-measure. Yang et al. [12] explore different methods of document vectorization (LDA, LSA, word2Vec, and doc2Vec), and a measure (TF-IDF) used to determine document similarity. For every document, the similarity is calculated using vector similarity metrics, such as cosine and KL-divergence. The models are evaluated using a dataset labeled by an expert, or an accuracy based on the total number of correctly retrieved citations in Wikipedia articles. In [13], the authors present a comparison between Continuous bag of words, Skip gram, Glove (Global Vectors for word representation) and the Hellinger-PCA (Principal Component Analysis) embedding models. These models are tested using the size of training data, the relation of the context and the target words, the memory consumption, the classifier used, and the effect of changes in the dimensionality of the model.

In [14], they use a Doc2Vec model in a corpus constructed with 7000 Bengali sentences, to analyze its feasibility in the Bengali sentiment analysis. The corpus consists of two types of data differentiated by their polarity, i.e., positive and negative. Then, they use several machine learning algorithms for comparing the accuracy of the classification. In general, the Bi-Directional Long Short-Term Memory (BLSTM) obtains the best results. Imaduddin et al. [15] use hotel review data obtained from the Traveloka website, to carry out sentiment analysis. The authors compare the performance of the following word embedding techniques: Word2Vec Continuous Bag of Words (CBOW), Word2Vec skip-gram, Doc2Vec, and Glove. In their experiments, Glove method has the highest accuracy, and Word2Vec skip-gram model has the lowest accuracy. In the work [16], the authors propose an approach of sentiment analysis based on term extraction using various text embedding methods. They use versions of the long short-term memory (LSTM) artificial neural network, extended with the conditional random field (CRF). They analyze the influence on performance of extending the word vectorization step with character embedding. They test their approach on the SemEval dataset.

According to their results, the bi-directional LSTM, or LSTM extended with CRF layer, outperforms regular LSTM. In general, they determine that word embedding affects the detection performance.

Some works have proposed approaches for text classification. The authors of [17] have proposed a text representation matrix, combining Word2Vec and LDA. This combination of word meaning and semantic features, is used by the LSTM neural network for text classification. The results of the LSTM classification model are better than the traditional machine learning models. The paper [18] presents a comparison of different text classification techniques for an automated semantic annotation, based on K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Naive Bayes, using either full-text or only the title of documents. The performance of the classifications on three datasets, using only titles, reaches the best results of quality, compared to the performance when using the full-text. In [19], Wei et al. proposed a model for learning generic text embedding, which can be used to learn short text representations. The model consists of two convolutional neural networks: one for extracting the semantic representations of short texts, and the other for learning the classification of short texts. They assume that the approximation of the semantic representations of short text is Gaussian, in order to minimize the KL-divergence to map semantic representations into low-dimensional spaces with Gaussian distributions. They test their approach on a Chinese text classification dataset.

2.2. Comparison with Previous Works

Table 1 shows a comparison of our approach with previous works. The criteria of comparison are: (a) Do they consider different datasets? (b) Have they been tested for the generation of features? (c) Have they been tested as feature extraction methods? (d) Do they use non-supervised metrics to evaluate the performance? (e) Is the work in the context of digital educational contents?

According to the Table 1, our work, ref [12,18] use different datasets. However, our approach uses non-labeled datasets: patents, journals and learning resources. In addition, it is the only one that uses learning resources and patents, and only another one uses a scientific publication dataset in its analysis [6]. In regard to the used techniques, our work is interested in feature extraction methods to transform text documents into a list of features that can be easily used and understood, like BM25 and TF-IDF, and methods of document vectorization to create numerical features using statistical analysis, like LDA, LSA, and Doc2Vec. The only other work that considers a mix of these techniques is [12]. Finally, most of previous papers used supervised metrics in order to test the quality of the methods, in contrast with our work where a different approach is presented using several types of unsupervised metrics: one based on information theory (entropy) and the others based on document similarity. Only [6] considers unsupervised metrics, and there are several works that consider document similarity, but none information theory metrics. Furthermore, in the context of digital educational contents, there are not many works. In [17] is used the THUCNews dataset, which contains 740,000 news divided in 14 categories, one of them is education news. The other is [20], which studies the scientific article recommendation problem. Our paper considers different types of education digital documents (learning resources, scientific publications, and patents), and uses the LOM standard for representing them. In this context, our paper selects some of the fields of the LOM metadata standard for being analyzed by the feature extraction methods. Finally, our paper used the Microsoft Research Paraphrase Corpus dataset, in order to analyse the behaviour of the techniques in a domain different to the educational context, for classification.

Table 1. Comparison with previous works.

Works	a	b	c	d	e
[5]		X			
[6]			X	X	
[16]		X			
[17]		X			X
[18]	X	X			
[19]		X			
[20]		X			X
[7]		X			
[21]		X			
[11]			X		
[14]		X			
[12]	X	X	X		
[8]		X			
[13]		X			
[15]		X			
[9]		X			
Our Approach	X	X	X	X	X

3. Feature Extraction Strategies

3.1. Based on Probabilities: LDA

Latent Dirichlet allocation (LDA) is a probabilistic model based on unsupervised learning, which supposes each document like a mix of topics, and each topic has a probability distribution over all words in the vocabulary [7,17]. The topic distribution reflects the overall semantic information of the text/document, expressed in the form of probability, which is the direct extraction of the deep features of the document.

LDA is based on the idea that each document contains several hidden topics, each of which contains a collection of words related to the topic [7,8]. LDA discovers the latent topics Z from a collection of documents D. For LDA, each document is a probability distribution over all words in the vocabulary. LDA model projects the documents in a topical embedding space, and generates a topic vector from a document, which can be used as the features of the document.

In this way, the LDA topic model defines two polynomial distributions [8]: the document-topic distribution (θ), and the word-vocabulary distribution (ϕ). The first represents the probability distribution of each topic in the document; and the other, the probability distribution of each word appearing in the topic. In addition, LDA model has three parameters [7,17]: α is the parameters of the Dirichlet distribution of the topic distribution in a document, β is the parameters of the Dirichlet distribution of the word distribution in a topic, and K represent the number of topics.

LDA requires a learning phase, in order to infer/discover θ and ϕ in documents, which can be used to predict any new document with a similar topic distribution. Methods as Gibbs' Sampling is used to generate distributions, assuming a Dirichlet prior for the distribution of words and topics within the document [17]. Different representations can be built since the documents, varying the amount of topics to be considered.

3.2. Based on Vector Analysis: LSA

Latent Semantic Analysis (LSA) is a distributional semantic technique, which is an extension of TF-IDF, to analyze the semantic relationship between a set of documents by using the term-document matrix and the singular value decomposition (SVD) [21], which are applied to the TF-IDF matrix. LSA returns a term-document matrix where similar documents and similar words are placed closer [21]. The specific number of columns in the output matrix is equivalent to the document topics. LSA can analyse linguistic properties as synonymy and polysemy of words.

3.3. Based on Deep Learning: Doc2Vec

Doc2Vec is an extension of Word2Vec, and it is embedded in Word2Vec. Word2Vec builds a distributed semantic representation of words in the document, such that it is trained in the context of each word, in order to build a predictive model [21].

Doc2Vec learns a conceptual representation of a document from a corpus of documents. This model learns to connect documents and words [12]. Thus, Doc2Vec tags the documents and uses them for the training phase. During the training of the model, it learns paragraph and word vectors that are a semantic representation of the documents. The paragraph and word vectors are averaged or concatenated, in order to represent each document [15].

This method is very generic and can be used to generate embeddings from documents of any length. Doc2Vec is based on a deep neural network, while previous methods are based on a representation of information learned from terms and documents [12,15,21]. The trained model can predict behavior of new documents. Furthermore, this technique can be used to predict a word given the other words in a document.

3.4. Based on Term Frequency: BM25

BM25 function is a ranking function that ranks a group of documents depend on the keywords that appear in each document. The BM25 function obtains the score for each (word, document) pair, in order to rank documents [11]. This function is a family of scoring functions. Traditionally, it has been used by search engines to rank correspondence between documents and search queries. Thus, the BM25 function is an information retrieval formula function, which belongs to the BM family of retrieval models, and determines the weight of a term t in a document d.

4. Evaluation

4.1. Experiments

This section presents experiments with the four techniques presented in Section 3, using three different types of contents: patents (PT), scientific publications (SP), learning objects (LO) and the Microsoft Research Paraphrase Corpus (MSRPC).

4.1.1. Experimental Protocol

Three datasets were used for testing and evaluating techniques presented in Section 3: one of patents (PT), another of scientific publications (SP) and the last one of learning objects (LO). They were obtained ad-hoc from online sources using different ways of acquisition.

PT was collected from the United States Patent and Trademark Office (http://patft.uspto.gov), using the query tool they have available online for obtaining full text from patents and scripts, for the automation of web requests and data acquisition.

SP was collected from the ScienceDirect repository (https://www.sciencedirect.com) making use of the API that Elsevier provides for researchers (https://dev.elsevier.com). Elsevier enables endpoints for different platforms like Scopus or ScienceDirect. In this last one, full text from publications can

be retrieved jointly with metadata information. A python script was used for the automation of the recollection of data.

LO was collected from Merlot repository (https://www.merlot.org/merlot). Merlot offers an API for querying metadata of learning objects, but most of the services are not for free. We were not provided with access to the API, neither for research purposes, so public available information of learning resources was collected using scrapping techniques with the selenium library in Python. In this investigation, we were only interested in descriptions, scrapping of public available data worked for us.

PT, SP, and LO datasets are composed of approximately 10.000 contents, the data used from these datasets is title, description and keywords (when available), as text input. Furthermore, MSRPC dataset has been used for evaluating paraphrase detection algorithms [22–30]. It consists of 5803 pairs of paraphrases extracted from web news pages, 4077 for training and 1726 for testing.

Each technique was trained independently with every type of content, in order to generate the features/descriptors for every single content. Then, these features/descriptors were evaluated using three metrics which are going to be explained later. Finally, the results and the comparisons are carried out.

The features are generated using the contents in the fields of the LOM standard like the title, the description, and have been filtered the texts in languages different than English.

A pre-processing step is used before entering the contents to the algorithms, the sequence is shown in Figure 1, and is as follows:

- **Concatenation:** Title, Description and Keywords (when available) of contents are concatenated in a single text line.
- **Tokenization:** Text data are separated into tokens using the word tokenizer from nltk (Python library).
- **Lower case:** Every token is converted to lower case, in order to recognize similar tokens like "Smith" and "smith" as only one.
- **Punctuation marks removal:** punctuation marks, such as ".", ",", ":", "!", etc., are removed from the text.
- **Stop words removal:** Words that are excessively frequent are removed from text, because it is known that they do not have significant information.
- **Lemmatization:** Tokens are converted to its lemma using the wordnet lemmatizer from nltk (Python library).

The resulting texts are analyzed by each technique. A Bayesian optimization meta-learning method is executed in a proper parameter space, to find out the optimal parameters for each technique.

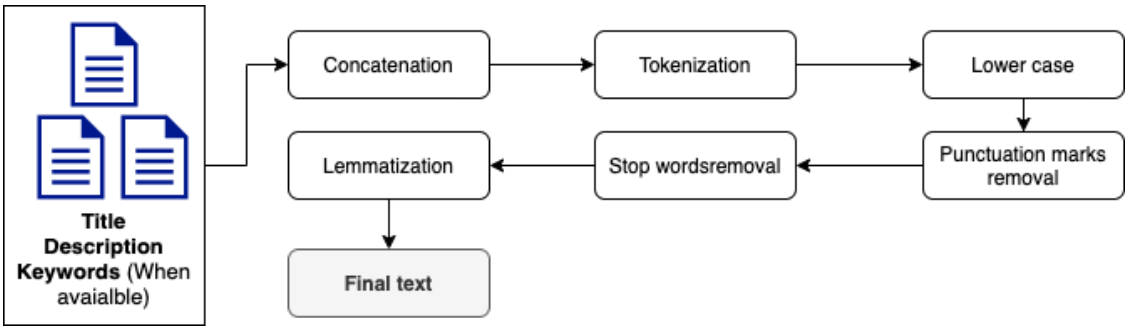

Figure 1. Text preprocessing.

4.1.2. Metrics

In order to compare the four techniques, three metrics for unsupervised contexts have been used. The first one is based on entropy, and the other ones based on similarity measures.

Metric Based on Entropy

Entropy is a measure that quantifies the average rate at which information is generated by a stochastic source of data. This entropy is known as the Shannon entropy. The intuition behind it, is the idea of measuring how much surprise there is at an event. Those events that are rare are more surprising, and therefore, have more information than those events that are common. So, those events with low probability have more information than those with high probability. In the clustering context, entropy associated with each possible cluster is the negative logarithm of the probability mass function for the cluster, and is computed as:

$$H = -\sum_i P_i \log P_i$$

For calculating this measure, we use k-means as clustering technique and the elbow method to determine the number of clusters. Thus, the Shannon entropy is computed for the clusters of the descriptors generated by each technique using k-means.

Metrics Based on Similarity Measures

A key concept behind document embeddings is their capacity to preserve semantic similarity in the descriptors' space; this idea is exploited for developing similarity measures to compare techniques of extraction of descriptors/features from texts.

Similarity between contents: is measured in two ways: semantic similarity between contents' text (similarity of texts), and similarity between contents' features (similarity of features). Similarity of texts is calculated based on [31]:

$$sim(T_1, T_2) = \frac{1}{2} * \left(\frac{\sum_{w \in \{T_1\}} maxSim(w, T_2) * idf(w)}{\sum_{w \in \{T_1\}} idf(w)} + \frac{\sum_{w \in \{T_2\}} maxSim(w, T_1) * idf(w)}{\sum_{w \in \{T_2\}} idf(w)} \right)$$

where T_n is the n-th document, $idf(w)$ is the inverse document frequency of word w and $maxSim(w, T_n)$ is the maximum similitude between word w and any word in T_n. The similitude between words is calculated using the Palmer similarity metric [32] with the WordNet taxonomy:

$$Sim(w_i, w_j) = \frac{2 * depth(LCS)}{depth(w_i) + depth(w_j)}$$

where w_n is the representation of the n-th word in the WordNet taxonomy, and LCS is the least common subsumer of both representations of the words in the WordNet taxonomy.

Mandala et al. shows some inconveniences that WordNet has [33], which were evidenced during the experiments. Because of this, sometimes semantic similarity could not be computed, so this was replaced for the cosine similarity between representations of words [34].

The second similarity metric of contents is determined using cosine similarity between the features of contents extracted by each technique. It is computed using the next formula:

$$Cos(A, B) = \frac{A \cdot B}{||A|| ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Thus, here we propose two measures: 1. correlation between the semantic similarity of the contents and similarity of features, and 2. coherence of the feature space.

Correlation between measures: is based on the idea that if two contents are semantically similar, then their descriptors should be similar. In other words, their representations in the descriptor space should be close to each other.

This measure is calculated computing the correlation of the similarity of texts and the similarity of descriptors/features. The correlation used is the Pearson correlation coefficient.

Coherence: it is based on the idea that the dispersion in the descriptor space should be similar to the dispersion in the content space. So, it calculates a similarity measure to pairs of generated content descriptors and a text similarity measure to the corresponding pairs of contents, then both similarities are compared.

For this measure, semantic similarity and an adaptation of the cosine similarity were used for the contents and the descriptors, respectively. The adaptation of the cosine similarity is:

$$Cos^*(A,B) = 1 + \frac{Cos(A,B) - 1}{2}$$

Standard deviation is used as a dispersion measure for the comparison of similarities.

4.1.3. Results

The techniques defined in section III are compared using correlation, coherence and entropy; Table 2 show the metrics for patents, scientific publications, and learning objects; respectively.

In general, from correlation metric, we can say that there is not evidence of a relationship between the similarity of the contents and the similarity of descriptors. For this metric, LSA works better than the other techniques for all datasets.

All techniques have coherence up to 0.5. So, we can say that the dispersion in the descriptor space is similar to the dispersion in the content space. There is no technique that overcomes the others on every dataset, but LDA is the best among them, in all cases, have a coherence over 80%.

Finally, the entropy of Doc2Vec and LDA work well for all datasets, with values over 0.89 and 0.83, respectively. Thus, the generated descriptors are representative of the contents.

Table 2. Evaluation of techniques per data set.

Content Type	Technique	Correlation	Coherence	Entropy
Patents	BM25	0.304	0.841	0.643
	Doc2Vec	0.281	0.735	0.923
	LDA	0.267	0.839	0.908
	LSA	0.364	0.510	0.745
Scientific Publications	BM25	0.121	0.630	0.404
	Doc2Vec	0.526	0.459	0.894
	LDA	0.186	0.816	0.838
	LSA	0.428	0.891	0.807
Learning Objects	BM25	0.261	0.534	0.337
	Doc2Vec	0.121	0.580	0.901
	LDA	0.305	0.878	0.865
	LSA	0.356	0.518	0.501

In general, there is not a technique that dominates by type of content. Using entropy criterion, Doc2Vec is the best technique while BM25 is the worst. However, LSA has acceptable result in terms of the coherence criterion, but LDA has very good results for all datasets.

So, LDA and Doc2vec get the best results for descriptors generation while BM25 gets the worst (only in one case it gives the best results). On the other hand, LSA does not behave as well as LDA or Doc2vec, but has good results for all metrics and datasets.

Now, MSRPC dataset is used, which has a binary output (is paraphrase or not), so we use precision, recall, and f1-score to evaluate the methods. For determining if two texts are paraphrase or not, cosine similarity is used between descriptors of both texts, if similarity is greater than 0.7 they are considered a paraphrase. The time for training each technique using this dataset is shown in Table 3.

Table 3. Time for training by technique on MSRPC dataset in seconds.

Technique	Time
BM25	7.814
LDA	16.374
LSA	11.625
Doc2Vec	143.085

The Table 4 shows the evaluation metrics for the MSRPC dataset. Doc2Vec has the best results in terms of recall and f1-score, while has the worst for precision. For BM25, LDA and LSA, f1-score and recall values are very similar. F1-score is the harmonic mean of both *precision* and *recall*. Then, Doc2Vec, which reaches the best f1-score, works very well for MSRPC dataset.

Table 4. Microsoft Research Paraphrase Corpus results.

Technique	Precision	Recall	f1_score
BM25	0.707	0.753	0.729
LDA	0.728	0.730	0.729
LSA	0.712	0.715	0.713
Doc2Vec	0.671	0.966	0.792

Table 5. Microsoft Research Paraphrase Corpus best results from other works.

Model	Precision	Recall	f1_score
Wan et al. [22]	0.735	0.680	0.690
Fernando and Stevenson [23]	0.752	0.913	0.824
Madnani et al. [24]	-	-	0.841
Segura-Olivares et al. [25]	0.684	0.979	0.806
Calvo et al. [26]	0.686	0.980	0.806
Kenter and De Rijke [27]	0.781	0.906	0.839
Lee and Cheah [28]	-	-	0.804
Lee and Cheah [29]	-	-	0.818
Lee and Cheah [30]	0.755	0.882	0.814

Table 5 shows reported metrics in literature for MSRPC dataset. In general, f1-score in these approaches reach values over 0.80, except for Wan et al. [22] that has the worst score. In our work, only Doc2Vec, with a f1-score of 0.792, can be compared with these results. So, Doc2Vec not only generate good features (entropy), but also works great for classification. As for the execution time, Doc2Vec has a huge execution time due to its learning phase.

The identification of paraphrases is a very relevant task for the purpose of this work, since it gives evidence of the amount of semantic similarity conserved when extracting descriptors from

texts using these techniques. Despite the main target with these techniques is not to find paraphrase, the results are not far from other works that particularly focuses on this task. Specially, doc2vec seem to be the winner technique from this challenge's perspective, due to the high performance scores with this technique to identify almost all true paraphrases. Its disadvantages are that it consider many no-paraphrases as if they were, and it has a very large execution time.

4.1.4. Time Complexity

Wu et al. [35] define the time complexity for LSA as $O(N^2 \times k^3)$, where N is the number of terms plus number of documents and k is the number of factors. For BM25, the time complexity is $O(m \times avgdl)$, where m is the number of documents and $avgdl$ is the average document length. Time complexity for LDA is $O(m \times n^2)$ if $m > n$, and $O(n^3)$ otherwise, where m is the number of documents and n is the number of features. Seifi and Ekhveh [36] shows that doc2Vec time complexity is very similar to Word2Vec one, adding the number of paragraphs in the training set to the vocabulary size. Doc2vec time complexity is: $e \times t \times (w \times n + n \times \log_2(v + p))$ where e is the number of training epochs, t is the number of words in the training set, w is the size of the input window, n is the size of the hidden layer, v is the size of the vocabulary of the training set, and p is the number of paragraphs in the training set.

Comparing the time complexity of the four models, we can say that BM25 has the best time, but the worst results, followed by LSA. LDA and Doc2Vec, which have the best results, in occasions have very large execution times. Table 3 shows evidence of this.

4.2. Use Cases

In this section, we consider two use cases. Again, the optimal parameters by technique are determined using a meta-learning approach for each use case.

4.2.1. Information Retrieval

For information retrieval use case, we implement document ranking with 2 datasets: Cranfield collection and Microsoft Machine Reading Comprehension.

Cranfield Collection

The Cranfield collection dataset [37] is available and distributed by University of Glasgow (Cranfield collection. http://ir.dcs.gla.ac.uk/resources/test_collections/cran). This dataset contains 325 queries with relevant documents per query, for a total of 1400 documents. Relevance in this dataset is measured from 1 to 5, where 1 is the maximum relevance and 5 is the minimum. For convenience, we invert the relevance scale and limit it to just 4 levels, so that 4 is the maximum relevance and 1 represents 4 from the original scale. So, the performance of the techniques is measured in two ways:

Let p_{ij} be the relevance points of the j-th document that was retrieved, and in fact, is relevant for the i-th query, p^*_{ij} the relevance points of the j-th relevant document for the i-th query, and q the number of queries.

The first metric is the mean of the sum of scores of retrieved documents that are relevant, divided by the total sum of relevance points of all relevant documents for the query, through all queries, calculated as follows:

$$score\% = \frac{\sum_{i=1}^{q} \frac{\sum_{j=1}^{g_i} |p^*_{ij}|}{\sum_{j=1}^{n_i} |p_{ij}|}}{q}$$

where g_i is the number of retrieved documents that are relevant for the i-th query, and n_i is the number of documents relevant for the i-th query.

The second metric is the mean of the quantity of retrieved documents that are relevant for each query, calculated as follows:

$$count\% = \frac{\sum_{i=1}^{q} \frac{g_i}{n_i}}{q}$$

We use all techniques to generate descriptors for document and query. The ranking was performed computing cosine similarity between features of documents and the specific query. Then, the predicted relevant documents are there ones that are over the 99th percentile of similarity, so only the most 14 (1% of 1400) similar document are predicted as relevant.

The results for this use case are shown in Table 6. BM25 gives the best values for the score and count metrics; however, the results of LSA are very close. For this experiment, the results are not good, they are inferior to 50%, even some techniques' score goes below 2% (Doc2Vec case). In general, these techniques have problems in retrieving relevant documents for the queries, and only BM25 and LSA have regular results.

Table 6. Techniques performance comparison with Cranfield dataset.

Technique	Score %	Count %
LSA	43.13	41.75
BM25	44.65	43.44
LDA	24.28	23.49
Doc2Vec	1.35	1.35

Microsoft Machine Reading Comprehension

The Microsoft Machine Reading Comprehension [38] is a public large scale dataset for non commercial uses that is available at MS Marco. This dataset contains more than 400 millions of pairs of queries, with relevant and non-relevant documents. In this case, we define two sets: the development and evaluation sets, and each one contains about 6900 queries with the top of 1000 most relevant documents per query. We use Mean Reciprocal Rank (MRR) as evaluation metric to be comparable with previous works [39–43]. A total of 100.000 documents are extracted from the total dataset for training the four techniques. Table 7 shows the training time for each technique.

Table 7. Time for training by technique on MS Marco dataset (100.000 training samples) in seconds.

Technique	Time
BM25	28.227
LDA	198.947
LSA	22.547
Doc2Vec	143.085

Table 8 gives evidence that these four techniques do not work well for document ranking comparing them with the state of the art (see Table 9). However, we compare them among themselves to give evidence of which technique is better. BM25 and Doc2Vec are the best technique for this task.

Table 8. Microsoft Machine Reading Comprehension results (MRR@10).

Technique	Dev	Eval
BM25	8.275	3.183
LDA	2.048	0.4561
LSA	5.548	3.022
Doc2Vec	6.395	6.453

Table 9. Microsoft Machine Reading Comprehension results (MRR@10) of other works.

Technique	Dev	Eval
Nogueira and Cho [39]	36.5	35.8
Mitra et al. [40]	33.3	-
Rosset et al. [41]	-	26.94
Nogueira et al. [42]	39.0	37.9
Padigela et al. [43]	35.87	36.53

4.2.2. Recommendation System

In this use case, a collection of 2860 course descriptions was extracted from online virtual learning platforms. Specifically, these course descriptions were extracted from Coursera (https://www.coursera.org) making use of web scrapping tools for collecting public available data about courses. The scrapping was performed using the selenium library in Python.

We use four techniques to generate descriptors for each course description, then a similarity measure is computed between the descriptors generated for course descriptions and for contents. The outputs of every technique are compared by type of content.

Each technique runs at least 10 times, top 10 recommended documents for each execution is saved, and then, the average is calculated per document, appearing at least once in the outputs of the run. LSA and BM25 are really stables talking about results, almost every execution outputs were the same documents with the same similarity, in 10 executions only 12 different documents appeared. Doc2Vec is a little more variable than these two techniques, in 10 executions 14 different documents appeared. LDA is not stable, in 10 executions 79 different documents appeared. Tables 10–12 show the top 5 recommendations for one of the courses.

We observe that there are few documents in common for various techniques, like in patents the documents 11358 and 7093, in scientific publication 3389, and in learning objects 1515.

In addition, BM25 gives pretty good results because the similarity measure among the course contents and any type of educational content is enough good, with and stable list of recommendations. In the case of LDA, in some cases gives very good values (for example, 79.8% for patents), but with a frequency of occurrence of the recommendation not very good (in the same example, 6 times).

Table 10. Top 5 recommended patents for the first course by technique.

Technique	Doc ID	Ocurrences	Mean Sim %	Max Sim %
BM25	5230	10	57.73	57.73
	11358	10	53.87	53.87
	2147	10	53.59	53.59
	4222	10	53.08	53.08
	7093	10	51.98	51.98
LSA	9475	10	40.01	40.56
	11358	10	38.38	39.36
	509	10	38.1	38.56
	10440	10	38.01	38.59
	7093	10	36.74	37.64
Doc2Vec	621	10	42.4	43.14
	7093	10	41.3	42.06
	11521	10	40.76	41.87
	4491	10	40.67	41.18
	11151	10	40.31	40.7
LDA	2923	6	79.8	81.59
	717	6	63.71	66.47
	5286	5	66.67	69.94
	6779	4	59.7	60.97
	9146	3	74.02	74.83

Table 11. Top 5 recommended scientific publications for the first course by technique.

Technique	Doc ID	Ocurrences	Mean Sim %	Max Sim %
BM25	6075	10	69.88	69.88
	1289	10	66.04	66.04
	4939	10	55.07	55.07
	3389	10	38.77	38.77
	2965	10	38.74	38.74
LSA	3020	10	66.41	67.03
	6072	10	65.43	66.0
	7062	10	59.31	59.79
	4916	10	59.06	59.89
	5952	10	54.89	55.61
Doc2Vec	6256	10	52.2	52.75
	1043	10	50.57	51.05
	6755	10	50.43	50.9
	3389	10	50.03	50.51
	949	10	48.11	48.77
LDA	4939	6	62.14	66.19
	6870	8	43.78	61.13
	6594	4	42.15	46.63
	4174	3	38.5	42.32
	5306	2	57.47	63.87

Now, we analyse the quality of the recommendations for the set of courses. We consider the average of the occurrences and the average of the similarity value, using a similarity threshold by type of content. The similarity thresholds are 40%, 60% and 80%.

Table 12. Top 5 recommended learning objects for the first course by technique.

Technique	Doc ID	Ocurrences	Mean Sim %	Max Sim %
BM25	6075	10	69.88	69.88
	1289	10	66.04	66.04
	4939	10	55.07	55.07
	3389	10	38.77	38.77
	2965	10	38.74	38.74
LSA	3020	10	66.41	67.03
	6072	10	65.43	66.0
	7062	10	59.31	59.79
	4916	10	59.06	59.89
	5952	10	54.89	55.61
Doc2Vec	6256	10	52.2	52.75
	1043	10	50.57	51.05
	6755	10	50.43	50.9
	3389	10	50.03	50.51
	949	10	48.11	48.77
LDA	4939	6	62.14	66.19
	6870	8	43.78	61.13
	6594	4	42.15	46.63
	4174	3	38.5	42.32
	5306	2	57.47	63.87

Table 13 shows the results for patents, where LDA gives a good recommendation (over 80%) and 13 not-so-good recommendations (over 60%). The other techniques require a threshold of 40% to obtain recommendations, particularly, LSA and Doc2Vec have very low similarity values.

Table 13. Number of recommended Patents with different % of similarity by technique, for the courses.

Technique	80%		60%		40%	
	count	mean %	count	mean %	count	mean %
BM25	0	0	0	0	10	51.94
LSA	0	0	0	0	1	40.56
Doc2Vec	0	0	0	0	7	41.39
LDA	1	81.9	13	68.53	50	61.08

Table 14 shows the results of scientific publications. In this case, there is not a technique that gives recommendations with 80% of similarity. BM25, LDA, and LSA give good results with a threshold of 60%.

Table 14. Number of recommended Scientific Papers with different % of similarity by technique, for the courses.

Technique	80%		60%		40%	
	count	mean %	count	mean %	count	mean %
BM25	0	0	2	67.96	3	63.66
LSA	0	0	2	66.51	10	55.81
Doc2Vec	0	0	0	0	12	49.14
LDA	0	0	2	63.66	5	56.03

Finally, Table 15 shows the results for learning objects. Again, there is no one technique that gives recommendations with 80% of similarity. In this case, LDA gives the best results, followed by BM25.

Table 15. Number of recommended Learning objects with different % of similarity by technique, for the courses.

Technique	80%		60%		40%	
	count	mean %	count	mean %	count	mean %
BM25	0	0	1	60.41	10	49.39
LSA	0	0	0	0	1	45.37
Doc2Vec	0	0	0	0	7	42.26
LDA	0	0	62	70.8	95	65.14

In general, LDA has the highest similarity measures between contents and courses. LSA and Doc2Vec perform poorly, particularly, in the case of patents and learning objects. On the other hand, BM25 recommends less contents, but normally they have a good similarity with the courses (superior than 50%).

4.3. Discussion of Results

The selected techniques do not preserve a similar behavior about the semantic similarity between the documents. Some techniques do not even have 20% of correlation with semantic similarity for some types of contents. There is a great opportunity for improvements in this field.

On the other hand, there is not a good technique for extracting descriptors from every kind of content. Each content type has a different best results' technique according to the metric used: BM25 for patents and coherence, or LSA for learning objects and correlation, or LDA for learning objects and coherence, or Doc2Vec for patents and entropy. In general, there is not a conclusion about what technique is better, it depends on data and metric used.

Doc2Vec technique gives the best results for the entropy metric, while LSA gives the worst ones. In general, LSA is the fastest technique and has the best results in terms of correlation, despite its low values. Nevertheless, LSA has the worst results for the coherence and entropy metrics.

It is observed that LDA works better with high number of clusters. It is possible that the datasets contain a lot of topics because have contents from diverse areas of knowledge, and this is causing the big quantity of clusters for this method. Coherence score is one of the metrics generally used for evaluating topic models. As expected, LDA shows a high coherence score in all cases, this because of its nature as topic detection model. Furthermore, the entropy of this model is always high, this can be understood as that there are dominant topics, so the probability distribution of the topics is not uniform or the document descriptors contain high information.

BM25 is similar to LSA, the same dimensionality reduction step is used in this case; however, results are different. In general, its entropy values are very bad, and for the rest of metrics, the results are very irregulars.

Finally, Doc2Vec, in general, is not quite far from best results, and shows the best correlation for scientific publications that indicates a good degree of semantic similarity relation between descriptors of contents. In addition, it shows a surprisingly high entropy for any content type, which indicates that it generates discriminant descriptors. It is seen that this technique works better with little windows, low learning rate and a high quantity of iterations.

On the other hand, in the context of classification problems, for the classification of paraphrases, the performance of our methods follows the best results reported in the literature. BM25, LDA, and LSA have very similar f1-score and recall values. However, Doc2Vec has the best results with respect to f1-score, but with a large execution time.

In document retrieval use case, LSA and BM25 outperforms LDA and Doc2Vec, and just for a little BM25 is over LSA in this task. Doc2Vec shows poor results in this scenario, and in general, for the different techniques, the relevant contents are not useful. Normally, the performances of the techniques are bad for document ranking.

In the case of recommendation system, LDA has a high variance (it generates a lot of recommendations and not always the same), but generates more relevant documents. For BM25, the results are quite steady and good. LSA and Doc2Vec give unpleasant results.

In general, the techniques do not have a pattern of optimal parameters, which requires to be tuned for every type of content. Finally, the execution times of Doc2Vec and LDA are substantial, and must be improved.

5. Conclusions

In this paper, we have carried out an in-deep evaluation of different approaches of feature extraction in the educational domain. We used techniques based on different models (BM25, LSA, Doc2Vec and LDA), and executed several trials on datasets with different characteristics, some were educational datasets (scientific articles, learning objects and patents) and others like the Microsoft Research Paraphrase Corpus. Additionally, we have defined unsupervised metrics and two uses cases, in order to perform the comparisons.

According to the results, there is not a unique technique that dominates the others, because each one has a better behavior for each type of content, or according to each use case. Moreover, their results are different according to the metric used. The metric of entropy measures the quality of features detected by the techniques (if they are discriminants), the correlation determines if the characteristics of the content space are kept in the feature space. Finally, the coherence determines the quality of the feature space created. Each technique exploits better one of these aspects, according to their theoretical bases.

Regarding to the experimental results of this paper, Doc2Vec is the best in the context of the entropy metric, and LSA for the case of correlation metric. However, the values for this last metric are poor, future works must analyse how to improve these results. For the classification problems of paraphrases, the performance of our methods follows the best results reported in the literature. In the document retrieval use case, the results are not as expected. Nevertheless, LSA and BM25 are the most notable methods. In general, the four techniques do not work well for document ranking, comparing them with the state of the art. For recommendation system, BM25 gives the more stable and better results, which are not bad.

In general, it is possible to analyse the theoretical formulation of each technique for each context of application, in order to define specific improvement strategies. It will be studied in future works. Furthermore, a further work must analyze other extraction methods, like Word2Vec, or TF-IDF or Lambda (a fuzzy clustering algorithm [44,45]). Finally, another work must analyze the behavior of the methods in a real VLE, considering metrics that evaluate the impact of the recommendation or the information recovered in the learning process (student score, etc.).

Author Contributions: Conceptualization, J.A., J.M.-P. and E.M.; methodology, J.A. and J.M.-P.; validation, C.S. and H.V.; formal analysis, J.A., C.S. and H.V.; investigation, J.A. and J.M.-P.; data curation, C.S. and H.V.; writing–original draft preparation, J.A., C.S., H.V. and J.M.-P.; writing–review and editing, J.A. and E.M.; supervision, J.A. and E.M.; project administration, E.M.; funding acquisition, E.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MinCiencias–Government of Antioquia–Republic of Colombia grant number 80740-019-2019: "Contenidos de aprendizaje inteligentes a través del uso de herramientas de Big Data, Analítica Avanzada e IA." and "The APC was funded by MinCiencias–Government of Antioquia–Republic of Colombia".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vargo, J.; Nesbit, J.C.; Belfer, K.; Archambault, A. Learning object evaluation: Computer-mediated collaboration and inter-rater reliability. *Int. J. Comput. Appl.* **2003**, *25*, 198–205. [CrossRef]
2. Pacheco, F.; Exposito, E.; Aguilar, J.; Gineste, M.; Baudoin, C. A novel statistical based feature extraction approach for the inner-class feature estimation using linear regression. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [CrossRef]
3. Rodriguez, T.; Aguilar, J. Knowledge Extraction System from Unstructured Documents. *IEEE Latin Am. Trans.* **2018**, *16*, 639–646. [CrossRef]
4. Learning Technology Standards Committeeof the IEEE. IEEE P1484.12.2/D1. Final Standard for Learning Technology—Learning Object Metadata. 2002. Available online: http://www.dia.uniroma3.it/~sciarro/e-learning/LOM_1484_12_1_v1_Final_Draft.pdf (accessed on 26 March 2020). [CrossRef]
5. Fano, E.; Karlgren, J.; Nivre, J. Uppsala University and Gavagai at CLEF Erisk: Comparing word embedding models. In Proceedings of the Working Notes of CLEF 2019 Conference and Labs of the Evaluation Forum (CLEF 2019), Lugano, Switzerland, 9–12 September 2019; Volume 2380.
6. Singh, A.K.; Shashi, M. Vectorization of Text Documents for Identifying Unifiable News Articles. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*. [CrossRef]
7. Peng, H.; Wang, J.; Shen, Q. Improving Text Models with Latent Feature Vector Representations. In Proceedings of the 2019 IEEE 13th International Conference on Semantic Computing (ICSC), Newport Beach, CA, USA, 30 January–1 February 2019; pp. 154–157. [CrossRef]
8. Niu, L.; Dai, X.; Zhang, J.; Chen, J. Topic2Vec: Learning distributed representations of topics. In Proceedings of the 2015 International Conference on Asian Language Processing (IALP), Suzhou, China, 24–25 October 2015; pp. 193–196. [CrossRef]
9. Ritu, Z.S.; Nowshin, N.; Nahid, M.M.H.; Ismail, S. Performance Analysis of Different Word Embedding Models on Bangla Language. In Proceedings of the 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, Bangladesh, 21–22 September 2018; pp. 1–5. [CrossRef]
10. Gorro, K.; Ancheta, J.R.; Capao, K.; Oco, N.; Roxas, R.E.; Sabellano, M.J.; Nonnecke, B.; Mohanty, S.; Crittenden, C.; Goldberg, K. Qualitative data analysis of disaster risk reduction suggestions assisted by topic modeling and word2vec. In Proceedings of the 2017 International Conference on Asian Language Processing (IALP), Singapore, 5–7 December 2017; pp. 293–297. [CrossRef]
11. Kadhim, A.I. Term Weighting for Feature Extraction on Twitter: A Comparison Between BM25 and TF-IDF. In Proceedings of the 2019 International Conference on Advanced Science and Engineering (ICOASE), Duhok, Iraq, 2–4 April 2019; pp. 124–128. [CrossRef]
12. Yang, J.; Ward, J.; Gharavi, E.; Dawson, J.; Alvarado, R. Bi-directional Relevance Matching between Medical Corpora. In Proceedings of the 2019 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 26 April 2019; pp. 1–6. [CrossRef]
13. Bhoir, S.; Ghorpade, T.; Mane, V. Comparative analysis of different word embedding models. In Proceedings of the 2017 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, 1–2 December 2017; pp. 1–4. [CrossRef]
14. Hoque, M.T.; Islam, A.; Ahmed, E.; Mamun, K.A.; Huda, M.N. Analyzing Performance of Different Machine Learning Approaches With Doc2vec for Classifying Sentiment of Bengali Natural Language. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, 7–9 February 2019; pp. 1–5. [CrossRef]
15. Imaduddin, H.; Widyawan.; Fauziati, S. Word Embedding Comparison for Indonesian Language Sentiment Analysis. In Proceedings of the 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), Yogyakarta, Indonesia, 13–15 March 2019; pp. 426–430. [CrossRef]
16. Augustyniak, Ł.; Kajdanowicz, T.; Kazienko, P. Comprehensive Analysis of Aspect Term Extraction Methods using Various Text Embeddings. *arXiv* **2019**, arXiv:cs.CL/1909.04917.
17. Liang, Q.; Wu, P.; Huang, C. An Efficient Method for Text Classification Task. In Proceedings of the 2019 International Conference on Big Data Engineering, Hong Kong, 11–13 June 2019; pp. 92–97. [CrossRef]
18. Galke, L.; Mai, F.; Schelten, A.; Brunsch, D.; Scherp, A. Comparing Titles vs. Full-text for Multi-Label Classification of Scientific Papers and News Articles. *arXiv* **2017**, arXiv:1705.05311.

19. Wei, Y.; Wei, J.; Yang, Z. Unsupervised learning of semantic representation for documents with the law of total probability. *Nat. Lang. Eng.* **2018**, *24*, 491–522. [CrossRef]
20. Gupta, S.; Varma, V. Scientific Article Recommendation by Using Distributed Representations of Text and Graph. In Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, 3–7 April 2017; pp. 1267–1268. [CrossRef]
21. Nandi, R.N.; Zaman, M.A.; Al Muntasir, T.; Sumit, S.H.; Sourov, T.; Rahman, M.J.U. Bangla News Recommendation Using doc2vec. In Proceedings of the 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, Bangladesh, 21–22 September 2018; pp. 1–5. [CrossRef]
22. Wan, S.; Dras, M.; Dale, R.; Paris, C. Using dependency-based features to take the'para-farce'out of paraphrase. In Proceedings of the Australasian Language Technology Workshop 2006, Sydney, Australia, 30 November–1 December 2006; pp. 131–138.
23. Fernando, S.; Stevenson, M. A semantic similarity approach to paraphrase detection. Available online: https://www.researchgate.net/profile/Samuel_Fernando/publication/228616213_A_Semantic_Similarity_Approach_to_Paraphrase_Detection/links/02e7e5204b323983fb000000/A-Semantic-Similarity-Approach-to-Paraphrase-Detection.pdf (accessed on 15 April 2020).
24. Madnani, N.; Tetreault, J.; Chodorow, M. Re-examining machine translation metrics for paraphrase identification. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Montreal, QC, Canada, 3–8 June 2012; pp. 182–190.
25. Segura-Olivares, A.; García, A.; Calvo, H. Feature Analysis for Paraphrase Recognition and Textual Entailment. *Res. Comput. Sci.* **2013**, *70*, 119–144. [CrossRef]
26. Calvo, H.; Segura-Olivares, A.; García, A. Dependency vs. constituent based syntactic n-grams in text similarity measures for paraphrase recognition. *Comput. Sist.* **2014**, *18*, 517–554. [CrossRef]
27. Kenter, T.; De Rijke, M. Short text similarity with word embeddings. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015; pp. 1411–1420.
28. Lee, J.; Cheah, Y.N. Semantic Relatedness Measure for Identifying Relevant Answers in Online Community Question Answering Services. In Proceedings of the 9th International Conference on IT in Asia (CITA), Kuching, Sarawak Malaysia, 4–5 August 2015.
29. Lee, J.C.; Cheah, Y.N. Paraphrase detection using semantic relatedness based on Synset Shortest Path in WordNet. In Proceedings of the 2016 International Conference On Advanced Informatics: Concepts, Theory and Application (ICAICTA), George Town, Malaysia, 16–19 August 2016; pp. 1–5.
30. Mahajan, R.S.; Zaveri, M.A. Modeling Paraphrase Identification Using Supervised Learning Methods Against Various Datasets and Features. In Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, 14–16 December 2017; pp. 1–4.
31. Mihalcea, R.; Corley, C.; Strapparava, C. Corpus-based and knowledge-based measures of text semantic similarity. In Proceedings of the National Conference on Artificial Intelligence, Boston, MA, USA, 16–20 July 2006; Volume 1, pp. 775–780.
32. Wu, Z.; Palmer, M. Verbs semantics and lexical selection. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, Las Cruces, NM, USA; 27–30 June 1994; pp. 133–138.
33. Mandala, R.; Takenobu, T.; Hozumi, T. The use of WordNet in information retrieval. In Proceedings of the Workshop Usage of WordNet in Natural Language Processing Systems, Montreal, QC, Canada, 16 August 1998.
34. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
35. Wu, C.; Potdar, V.; Chang, E. Latent semantic analysis–the dynamics of semantics web services discovery. In *Advances in Web Semantics I*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 346–373.
36. Seifi, S.T.; Ekhveh, A.A. Representing Unequal Data Series in Vector Space with Its Application in Bank Customer Clustering. In Proceedings of the International Congress on High-Performance Computing and Big Data Analysis, Tehran, Iran, 23–25 April 2019; pp. 315–330.

37. Cleverdon, C. The Cranfield tests on index language devices. In *Aslib Proceedings*; MCB UP Ltd.: Bingley, UK, 1967.
38. Bajaj, P.; Campos, D.; Craswell, N.; Deng, L.; Gao, J.; Liu, X.; Majumder, R.; McNamara, A.; Mitra, B.; Nguyen, T.; et al. MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv* **2016**, arXiv:1611.09268.
39. Nogueira, R.; Cho, K. Passage Re-ranking with BERT. *arXiv* **2019**, arXiv:1901.04085.
40. Mitra, B.; Rosset, C.; Hawking, D.; Craswell, N.; Diaz, F.; Yilmaz, E. Incorporating query term independence assumption for efficient retrieval and ranking using deep neural networks. *arXiv* **2019**, arXiv:1907.03693.
41. Rosset, C.; Mitra, B.; Xiong, C.; Craswell, N.; Song, X.; Tiwary, S. An Axiomatic Approach to Regularizing Neural Ranking Models. *arXiv* **2019**, arXiv:1904.06808.
42. Nogueira, R.; Yang, W.; Cho, K.; Lin, J. Multi-stage document ranking with BERT. *arXiv* **2019**, arXiv:1910.14424.
43. Padigela, H.; Zamani, H.; Croft, W.B. Investigating the Successes and Failures of BERT for Passage Re-Ranking. *arXiv* **2019**, arXiv:1905.01758.
44. Morales, L.; Lozada, H.; Aguilar, J.; Camargo, E. Applicability of LAMDA as classification model in the oil production. *Artif. Intell. Rev.* **2019**, *53*, 2207–2236. [CrossRef]
45. Waissman, J.; Sarrate, R.; Escobet, T.; Aguilar, J.; Dahhou, B. Wastewater treatment process supervision by means of a fuzzy automaton model. In Proceedings of the 2000 IEEE International Symposium on Intelligent Control, Rio Patras, Greece, 19 July 2000; pp. 163–168.