IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 19, 2020, accepted November 29, 2020, date of publication December 2, 2020,
date of current version December 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3042099

Deep-Learning Approach to Educational Text Mining and Application to the Analysis of Topics' Difficulty

LOURDES ARAUJO, FERNANDO LÓPEZ-OSTENERO, JUAN MARTÍNEZ-ROMO, AND LAURA PLAZA

Departamento de Lenguajes y Sistemas Informáticos, Universidad Nacional de Educación a Distancia (UNED), 28040 Madrid, Spain

Corresponding author: Lourdes Araujo (lurdes@lsi.uned.es)

This work was supported in part by the Universidad Nacional de Educación a Distancia (UNED) under Grant GID2017-1(2019).

ABSTRACT Learning analytics has emerged as a promising tool for optimizing the learning experience and results, especially in online educational environments. An important challenge in this area is identifying the most difficult topics for students in a subject, which is of great use to improve the quality of teaching by devoting more effort to those topics of greater difficulty, assigning them more time, resources and materials. We have approached the problem by means of natural language processing techniques. In particular, we propose a solution based on a deep learning model that automatically extracts the main topics that are covered in educational documents. This model is next applied to the problem of identifying the most difficult topics for students in a subject related to the study of algorithms and data structures in a Computer Science degree. Our results show that our topic identification model presents very high accuracy (around 90 percent) and may be efficiently used in learning analytics applications, such as the identification and understanding of what makes the learning of a subject difficult. An exhaustive analysis of the case study has also revealed that there are indeed topics that are consistently more difficult for most students, and also that the perception of difficulty in students and teachers does not always coincide with the actual difficulty indicated by the data, preventing to pay adequate attention to the most challenging topics.

INDEX TERMS Deep learning, text mining, learning analytics, teaching of algorithms, challenging topics.

I. INTRODUCTION

The widespread use of computers and the Internet in all areas of education has led to the availability of large amounts of data. The use of these data requires, on the one hand, the application of data mining techniques, usually known as educational data mining, and on the other hand, the development of techniques to analyze these data and apply them to improve the learning process, which is often referred to as learning analytics. Both fields can be seen as expressions of the same area of research in education.

Educational data mining and learning analytics have been considered in a large amount of works and applications [2], [22], [23], leading to a new generation of learning tools and educational paradigms, such as collaborative learning [27], personalized learning [11], blended learning [25], or game learning [3], to cite a few of them.

The associate editor coordinating the review of this manuscript and approving it for publication was Ravinesh C. Deo

Text is one of the main ways of information transmission and interaction in education. Text mining is hence essential to take advantage of this huge source of information. Natural language processing and machine learning techniques are applied to extract the data contained in documents. They make possible applications such as selecting documents in a personalized way to meet learning needs, creating question-answering systems, recommend content to enhance learning, etc. These techniques have been often used to analyze students' online communications, such as discussion forum [17], [24] allowing to capture social aspects of student learning. Sentiment analysis [1], [4], [21] has also been frequently employed to personalize recommendations in education considering both lexicon-based [4] and machine learning [21] approaches.

In this work, we apply natural language processing and text mining techniques to identify the main topics of a subject that are covered in a text. This characterization has many applications in education. It can be used, for example, to

identify the topics in the students' questions and provide automatic answers and recommendations. It can also be used in combination with sentiment analysis techniques to detect whether there is a particularly problematic issue. Another application may be the automatic tagging of videos from the transcripts. Therefore, the **first contribution** of this work is the design and implementation of a deep learning model that considers both lexical and syntactic features and that provides very accurate results (around 90 percent accuracy) in the identification of the main topics that are covered in a text. The problem is addressed as a binary classification task for each topic in a predefined list. Each classifier is implemented by a deep learning model using a Long Short-Term Memory (LSTM) neural network. The main input to the classifier are word embeddings, a distributed representation of words such that words with similar meaning have a similar representation, a key feature for the outstanding performance of deep learning methods in NLP tasks. We have also investigated the contribution of other features such as the part of speech of the words in the text or the presence of uppercase letters.

The **second and most important contribution** of this work is the application of the topic identification model to automate the process of understanding what makes the learning of a subject difficult. This application is of great use to improve the quality of teaching by devoting more effort to those topics of greater difficulty, assigning them more time, resources and materials. First, the topic detection model is applied to the automatic identification of the topics covered in the questions from a repository of past tests, and second, each topic is related with the performance of the students in the questions that deal with such topic. This makes possible to study whether there are significant differences between the students' scores for the different topics in the subject, or the differences are punctual and disappear in the aggregated data. We may also analyze the degree of confidence of the results and their correlation with the practical or theoretical nature of the questions.

As the **third relevant contribution**, we apply our methodology to a subject related to the study of algorithms and advanced data structures. This is a key field in Computer Science degrees and one of the most difficult, so that improvements in this subject may have a great impact on the students' perception and satisfaction. Since the results have confirmed the existence of significant differences, we have investigated the reasons for the differences in difficulty. Moreover, a study of the perception of students and teachers carried out by means of a questionnaire has revealed that the perceived difficulty does not always coincide with the difficulty according to the evaluation data. For example, both groups, students and teachers, do not consider divide and conquer scheme as the most difficult algorithmic scheme, whereas the data show that this is worst performance scheme. The questionnaire has also indicated possible reasons for the difficulty of some topics, such as the lack of sufficient prior knowledge about recursion, or the wrong perception of the topic difficulty, which leads students to devote less time than necessary to its study.

In summary, the main contributions of this work are:

- The design and implementation of a new deep learning model able to label the main topics that are covered in educational texts with very high accuracy, including a comprehensive evaluation of different types of embeddings for the representation of texts.
- The use of the topic labeling model in a novel application with important implications in education: the identification, based on objective data, of the most difficult topics for students in a subject. The automation of the topic identification process will allow for a quick analysis of student results for each topic, providing relevant information for teachers to improve materials and resources according to the students' needs.
- The application of the methodology for analyzing the topics' difficulty to a particular case study: a Computer Science subject related to the study of data structures and algorithmic schemes. This study has yielded important findings, such as the subjectivity of students and teachers' perception on the difficulty of the different topics and the need for an objective analysis based on data.

The remaining of the paper is organized as follows: section II presents the experimental framework used to obtain the experimental data and the deep learning model for topic classification; section III describes the results obtained by the classifiers when tagging the exam questions of the considered subject, section IV is devoted to present an application of the proposed labeling algorithm to analyze the difficulty of the topics in the subject considered, including data about the subjective estimation of topic difficulty according to students and teachers; and finally, section V draws the main conclusions and future work.

II. MATERIALS AND METHODS

This section presents the process followed to gather and annotate the data needed for training and evaluating our proposal, as well as the deep learning LSTM network for topic identification.

A. DATA PREPARATION

Our proposal has been tested on the data collected over the years on a subject related to algorithms and advanced data structures. The subject considered is taught at one of the largest Spanish Universities (in number of students) that combines distance learning with on-site assistance to students. Our study involves 2043 students who took the Algorithm and Data Structures tests from 2012 to 2020.

The subject is taught in the second course of a Computer Science degree and includes a part devoted to data structures and another dedicated to algorithmic schemes. Among the data structures that are taught are *hash tables*, *graphs* and *heaps*. These two last are used in the implementation of some algorithmic schemes that require to explore the search space (*graphs*) or the use of a priority queue (that can be implemented using *heaps*) for the *branch and bound* scheme.

Among the algorithmic schemes included in the subject are *greedy*, *divide and conquer*, *dynamic programming*, *backtracking*, and *branch and bound*.

For teaching each scheme and data structure, the general case is first presented and exemplified in a particular problem. Then, other classic problems of application of the data structure or the scheme are shown. About ten examples of algorithms are proposed for each scheme, including the most representative cases. For example, the *Prim*, *Kruskal*, and *Dijkstra* algorithms are studied in the *greedy scheme*, *quicksort* and *mergesort* in *divide and conquer*, and *Floyd* algorithm in *dynamic programming*. Other problems are also proposed for each topic to allow students to practice. Students also have to carry out two compulsory practical assignments in which they apply two of the schemes included in the subject, which are different for each academic year.

The tests of the subject are composed of multiple choice questions. Each question has four options, and students must choose one, and only one, option, or leave the question unanswered (blank). Wrong answers have a penalty.

The first step of the methodology consists in defining a set of labels or descriptors and assigning them to the questions in the tests of previous courses. Figure 1 shows the hierarchy of labels assigned and their meaning. Among the labels considered are the topics from the subject to which the question is related, the theoretical or practical nature of the question, and the previous knowledge required to understand the topic. Note that some labels may contain other more specific labels, such as, for instance, the label SCHEME, which comprises five specific schemes under it.

The second step is annotating the questions of the tests with the selected labels. As a result, our exam database consists in 26 different tests and 156 questions collected over a eight-year period.

HIERARCHY OF LABELS
- **PRACTICAL:** Question of a practical matter
- **THEORETICAL:** Question of a theoretical nature
- **COST:** Question related to Algorithmic cost
- **DS:** Question related to Data Structures
  - **HEAPS:** Question related to Heaps
  - **GRAPHS:** Question related to Graphs
  - **HASH:** Question related to Hash Tables
- **SCHEME:** Question related to Algorithmic Schemes
  - **GREEDY:** Question related to Greedy Scheme
    - **TASK_SCHE:** Greedy algorithm for scheduling
    - **PRIM:** Prim algorithm
    - **KRUSKAL:** Kruskal algorithm
    - **DIJKSTRA:** Dijkstra algorithm
  - **D&C:** Question related to Divide and Conquer Scheme
    - **QUICKSORT:** Quicksort algorithm
  - **DP:** Question related to Dynamic Programming Scheme
  - **BA:** Question related to Backtracking Scheme
  - **B&B:** Question related to Branch and Bound Scheme

FIGURE 1. Hierarchy of labels along with their meaning, assigned to topics and aspects of the considered field.

B. DEEP LEARNING MODEL FOR THE BINARY CLASSIFICATION OF TOPIC LABELS

The annotated dataset allows the development of automatic labeling systems to classify educational texts according to the main topic they deal with. In this case, texts are related to algorithms and advanced data structures. In this section, we present the design of the deep learning neural network model for performing this task using the annotated dataset for training and evaluation.

The problem is addressed as a binary classification for each topic in the subject. The size of the training data is not large enough to consider other kind of classifiers. Each test question in the training data is assigned a label TRUE if it is related to the topic being classified and FALSE otherwise.

The model proposed is based on a Bi-LSTM network [16]. LSTM networks are able to learn long-term dependencies and to remember information for long sequences of input. As any other recurrent neural network, they are a chain of repeating modules of neural networks. Bi-LSTMs are bidirectional LSTMs, which connect two hidden layers of opposite directions to the output. In this way the output layer gets information from both, backwards and forward states simultaneously. A *densely* connected hidden layer (Dense), with sigmoid activation function, takes the Bi-LSTM output as input and provides the final probabilities for the document to be associated to the considered label.

The exam questions are collected in xml files, such as the one shown below.

```
<TEST date="2012-F-1S">
  <QUESTION n="2">
    <TEXT> Consider the problem of the backpack problem...
    </TEXT>
    <OPTION l="a" v="F">0 2 2 5 10 12 ...</OPTION>
    <OPTION l="b" v="T">0 2 2 5 10 14 ...</OPTION>
    <OPTION l="c" v="F">0 2 2 5 10 12 ...</OPTION>
    <OPTION l="d" v="F">None of the above</OPTION>
  </QUESTION>
</TEST>
```

Each exam question in the training and validation sets is annotated with the labels associated to it, for example:

2012|FEB|1S|2|SCHEME,DP,PRACTICAL

For each label, all the exam questions are examined and labeled as positive if they have been assigned to a particular label, or as negative otherwise.

In order to design our deep learning model we have considered the following features:

Words. We have used the embedding vectors by Cardellino [8]. They are vectors of 300 dimensions corresponding to 1000653 unique tokens in Spanish. They have been obtained from a Spanish

billion words corpus using the word2vec [19] algorithm.

POS tagging. Another feature considered has been the POS tags assigned to the words in the document. The particular sequence of POS tags can help to select the correct labels. To assign the POS tag we have used the Spanish model of the open-source library Spacy for NLP in python, trained on the AnCora and WikiNER corpus.

Casing. Another interesting feature is the word casing information. A common NLP practice [12], that we have also applied here, is to reduce the number of entries in the dictionary by transforming all the words to lower case. However, capital letters may indicate the presence of some relevant entity names (Dijsktra, Prim, etc.) whose presence may be relevant for the topic being classified. Therefore, in order to keep some upper case information lost by the lower case transformation, we use the CASE embedding representation. Specifically, we use a CASE feature to indicate if a word is lowercase, is all uppercase, has first letter capital, or has at least one non-initial capital letter.

Chars. In order to improve the coverage of the word embeddings, for words, numbers, and other symbols not included in the tokens represented by the word embeddings, we resort to character embeddings. This vectorial representation captures the information contained in both prefixes and suffixes.

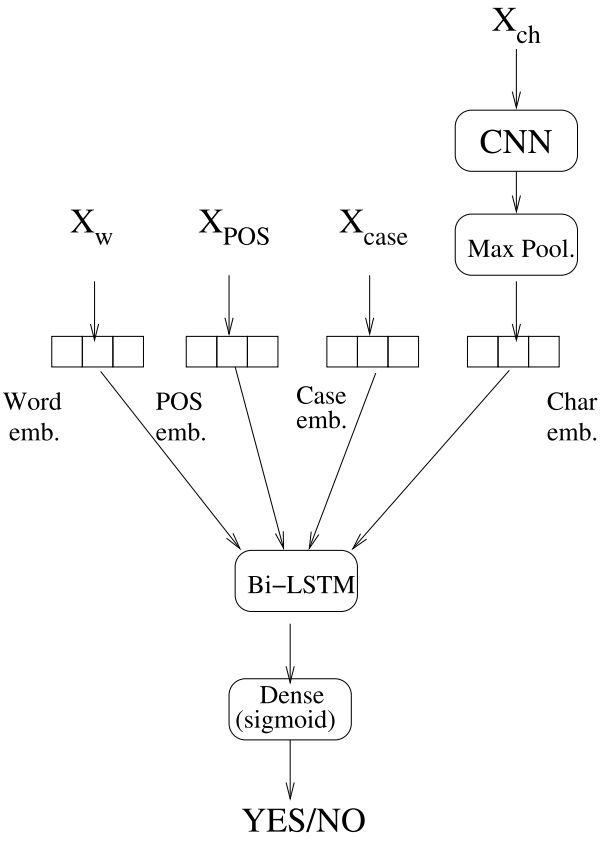


FIGURE 2. Deep learning model to classify questions according to each topic in the subject.

Figure 2 shows the architecture of the proposed model. The network is fed with four features represented by embeddings: lower case words, POS tagging, the casing information, and char information. A bi-directional LSTM layer, with hyperbolic tangent (tanh) activation function, takes the concatenation of the embeddings as input. LSTM are recurrent neural network (RNN) composed of Long short-term memory (LSTM) units. These units have the capability of ''remembering'' values over arbitrary time intervals, and therefore they are appropriate to process and predict series given by sequences of labels of unknown size. Finally, after another dense hidden layer, a dense layer, with sigmoid activation function, calculates the probabilities of the positive or negative answer of the classifier.

The PoS-tagging, casing and character embedding models have been implemented using Keras Embedding Layers initialized using a random uniform distribution. In the case of the char embeddings, we have also tested the system performance using pre-trained char embeddings. To generate these pre-trained embeddings we have used the architecture proposed by Kim *et al.* [18]. They proposed a neural language model that utilizes only character-level inputs where predictions are still made at the word-level. The authors claim that the model is able to encode, from characters only, rich semantic and orthographic features. We have applied the architecture proposed in this work to generate 25 dimensions char embeddings. The model has been trained with a collection of 100.000 documents collected from PUBMED. The python code is a available in the web page of our research group.[1]

III. RESULTS

We have split our collection of exam questions into training, validation and test sets. Specifically, the training set is composed of 15 different tests and 90 questions collected over a five-year period (February 2012–September 2016), the validation set is composed of the exams corresponding to 2017 and 2018, and test set is composed of 36 questions from 6 exams collected during the two last years (2019 and 2020).

For each configuration considered, we have performed a detailed study of the hyperparameters of the network using the validation data set: the Bi-LSTM size and the number of training steps or epochs. Figures 3 and 4 show the study of these parameters for the model using only word embeddings, whereas Figures 5 and 6 show the study for the configuration using word embeddings and pretrained word embeddings. A similar study has been performed for the rest of the configurations considered. Table 1 shows the parameters selected for each of them. In general, best results are obtained for not very large networks of between 10 and 25 LSTM units. This is

[1] http://ineda.lsi.uned.es/recursos/char_pretrained_model.tar.gz

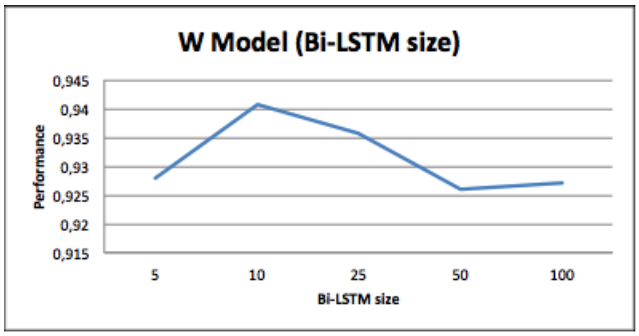


FIGURE 3. Study of the optimal Bi-LSTM size for the model using only word embeddings. Number of epoch is set to 100.

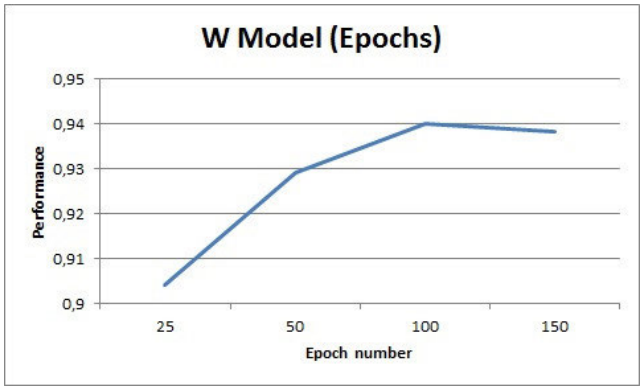


FIGURE 4. Study of the optimal number of training steps for the model using only word embeddings. Bi-LSTM size is set to 25.

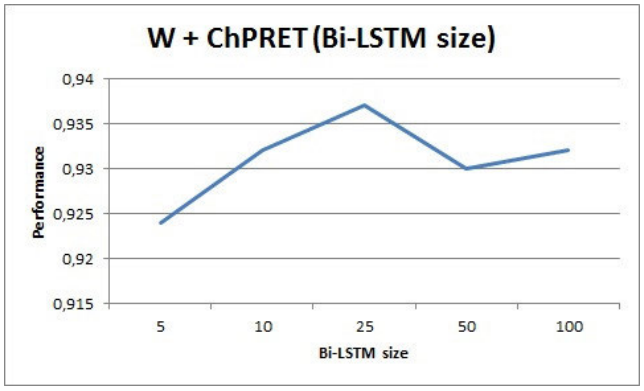


FIGURE 5. Study of the optimal Bi-LSTM size for the model using pretrained embeddings for words and chars. Number of epoch is set to 100.

consistent with the fact that the processed texts corresponding to exam questions are not very long. In all cases, 100 epochs are enough to achieve a stable result.

Table 2 shows the average results for all the topics considered using different configurations for the input. The first row shows the results using only the pre-trained word embeddings as input. The next three rows show the results obtained by enhancing the input with POS tag embeddings, casing embeddings and character embeddings, respectively.

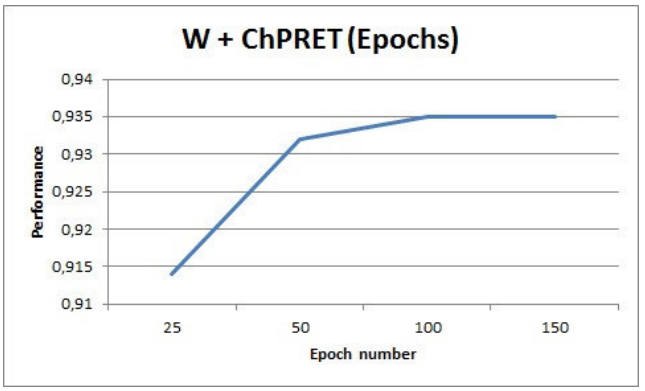


FIGURE 6. Study of the optimal number of training steps for the model using pretrained embeddings for words and chars. Bi-LSTM size is set to 10.

TABLE 1. Parameters selected for the different configurations of the network that have been considered.

Network conf.	Bi-LSTM size	epoch number
W	25	100
W+POS	10	100
W+Case	10	100
W+Ch	25	100
W+ChPRET	10	100
W+ChPRET+POS	25	100
W+ChPRET+CASE	25	100
W+ChPRET+POS+CASE	25	100

TABLE 2. Accuracy (average of 10 runs) results for different configurations of the input to deep learning LSTM classifier. Results are the average of the accuracy obtained for the set of classifiers for all the topic labels considered. Standard deviation appears in brackets. Parameters used for each configuration are shown in Table 1.

Features	Accuracy
W	0.930 (0.004)
W+POS	0.929 (0.001)
W+Case	0.933 (0.006)
W+Ch	0.925 (0.003)
W+ChPRET	0.934 (0.002)
W+ChPRET+POS	0.929 (0.003)
W+ChPRET+CASE	0.936 (0.005)
W+ChPRET+POS+CASE	0.935 (0.003)

The last four rows correspond to using pre-trained character embeddings obtained from a large dataset, as explained in the subsection II-B. We can see that the average results for all of experiments are high and similar. However, if we look at each of the features analyzed, we see that the use of POS tag embeddings slightly reduces the performance of pre-trained word embeddings and the character embeddings worsen the results a little more. This trend is confirmed by the use of pre-trained character embeddings obtained from PUBMED, where the best results are obtained with the use of casing embeddings. If we consider the accuracy of individual labels, shown in the Table 3, we can see that the results are particularly high for the most specific labels, even when using the simplest configuration (only word embeddings). In this way, for instance, we obtain nearly 100% accuracy for the HASH, DIJKSTRA, PRIM, QUICKSORT labels. However,

TABLE 3. Accuracy (average of 10 runs) results for each topic label using the configuration that has as input only word embeddings. Standard deviation appears in brackets.

Topic	Accuracy
DS	0.916(0.051)
HEAPS	0.95(0.012)
GRAPHS	0.905(0.015)
HASH	0.972(0)
SCHEME	0.872(0.024)
GREEDY	0.927(0.015)
TASK_SCHE	0.966(0.012)
PRIM	0.972(0)
KRUSKAL	0.944(0)
DIJKSTRA	1(0)
D&C	0.916(0)
QUICKSORT	0.972(0)
DP	0.888(0)
BA	0.961(0.015)
B&B	0.861(0)
PRACTICAL	0.938(0.023)
THEORETICAL	0.844(0.024)
COST	0.938(0.012)

the problem becomes more complex when considering labels at the highest level of the hierarchy, i.e. those representing general topics, such as THEORETICAL or SCHEME. The classification of these labels is more complex, since they are not generally described using specific words, and they involve different kinds of questions.

We have studied whether it is possible to improve the classification results of the most general labels by enriching the model with additional information. Table 4 shows the results for the different configurations considered. We can see that the differences are still small. In particular, we can see that the use of the POS embeddings, casing and char embeddings gets some improvement for some labels, but not for all of them. The greater improvement is obtained using pre-trained character embeddings. This indicates that the available data may not be enough to produce quality embeddings in some cases.

Considering the differences between the four top-level labels, we can observe that the highest improvement when using pre-trained character embeddings is achieved for the PRACTICAL label, as they capture the presence of numbers and other characters that may be indicative of the practical nature of a question. The same occurs for the DS label, the reason being that questions about data structures usually involve numbers. In contrast, the highest improvement for the SCHEME label is achieved when the case embeddings are used: it must be noted that the names of the different schemes, as well as the algorithms implementing them, are usually capitalized in Spanish. For this reason, the uppercase information is of great help.

Finally, it is worth mentioning the great advantage of the proposed system as it does not require complex engineering of features, as it happens with traditional machine learning systems. The system designed does not need any pre-processing of the data, but only low dimensional vectors representing the words and the sequences of characters. In order to support this claim, we have also tested some classic machine learning algorithms. Table 5 compares the results of some classic classifiers, SVM and Random Forest, and those obtained with the deep learning network proposed in this work.

For building the classic classifiers we have designed set of features adapted to the problem considered. These features, from the state of the art, include all the terms found in the labels considered (see Figure 1), as well as some other data, such as the number of letters, digits, and punctuation marks. We have also included TF-IDF (term frequency-inverse document frequency) features that indicates the relevance of terms included in the considered question with respect to the whole set of questions. We have used the Weka software[2] with default parameters for building these classifiers. We have evaluated our model using the same training and test sets that the deep learning system. Table 5 compares the results for a SVM classifier (without including the TF-IDF features and including them), and a Random Forest classifier, both with and without the TF-IDF features. For the sake of comparison, we have included, in the last row, the results for the best configuration of the deep learning system, which correspond to W+ChPRET+CASE in Table 4. We can observe that the classic classifiers can get close to the deep learning system results, but they require much more work for designing and extracting the used features. Actually, some features, such as TF-IDF, have a great influence on the results.

[2]https://www.cs.waikato.ac.nz/ml/weka/

TABLE 4. Accuracy (average of 10 runs) results for the more complex labels using different configurations of the inputs to the network: only word embedding (W), word and POS tag embeddings (W+POS), word + Case embedding (W+CASE), word and char embedding (W+CH), word and pre-trained char embeddings (W+ChPRET), word, pre-trained char and POS tag embeddings (W+ChPRET+POS), word, pre-trained char and casing embeddings (W+ChPRET+CASE), and finally word, pre-trained char, POS tag and casing embeddings (W+ChPRET+POS+CASE). Standard deviation appears in brackets. Best results appear in bold. Parameters used for each configuration are shown in Table 1.

Features	THEORETIC	PRACTICAL	DS	SCHEME
W	0.844(0.024)	0.938(0.023)	0.916(0.051)	0.872(0.024)
W+POS	0.850(0.024)	0.955(0.024)	0.911(0.023)	0.888(0.027)
W+Case	0.866(0.023)	0.933(0.024)	0.894(0.023)	**0.911(0.023)**
W+Ch	**0.900(0.015)**	0.950(0.012)	0.916(0.034)	0.883(0.030)
W+ChPRET	0.855(0.030)	**0.972(0)**	**0.961(0.015)**	0.861(0.027)
W+ChPRET+POS	0.838(0.030)	0.944(0.019)	0.950(0.012)	0.844(0.042)
W+ChPRET+CASE	0.850(0.031)	0.955(0.015)	**0.961(0.042)**	0.850(0.015)
W+ChPRET+POS+CASE	0.844(0.015)	**0.972(0)**	0.955(0.024)	0.866(0.012)

TABLE 5. Comparison of two classic classifiers, SVM and Random Forest, without including TF-IDF features (main features) and including them (w. tfidf). Last row corresponds to the results obtained with the best configuration of the deep learning (DL) system (W+ChPRET+CASE).

Classifier	Accuracy
SVM (main features)	0.872 (0.12)
SVM (w. tfidf)	0.910 (0.11)
R. Forest (main features)	0.899 (0.16)
R. Forest (w. tfidf)	0.9255 (0.14)
DL (best)	0.936 (0.005)

TABLE 6. Accuracy results for each topic label using the classification algorithm Random Forest in a machine learning approach. Mean absolute error appears in brackets.

Topic	Accuracy	Accuracy (w.o TFIDF)
DS	0.900(0.245)	0.933(0.216)
HEAPS	0.933(0.124)	0.933(0.095)
GRAPHS	0.900(0.168)	0.900(0.168)
HASH	0.933(0.109)	0.933(0.133)
SCHEME	0.966(0.144)	0.866(0.216)
GREEDY	0.866(0.175)	0.900(0.202)
TASK_SCHE	0.966(0.038)	0.966(0.022)
PRIM	0.966(0.050)	0.966(0.066)
KRUSKAL	0.900(0.094)	0.900(0.116)
DIJKSTRA	1.0(0.054)	0.933(0.138)
D&C	0.866(0.159)	0.900(0.122)
QUICKSORT	0.966(0.068)	0.933(0.088)
DP	0.833(0.171)	0.833(0.242)
BA	0.966(0.068)	1.00(0.044)
B&B	0.866(0.160)	0.833(0.161)
PRACTICAL	0.933(0.153)	0.733(0.255)
THEORETICAL	0.966(0.232)	0.800(0.245)
COST	0.933(0.124)	0.933(0.143)

Table 6 shows the results for the different topics using the Random Forest classifier, that provides better results than SVM. The results for nearly all labels are lower or similar to those obtained with the deep learning system (see Table 3). Only the results for the SCHEME label are slightly higher and only using TF-IDF features. The simplicity of the deep learning system is important in order to apply the system to different tasks and fields in education. In particular, the high results achieved validate the system to be applied to the labeling of topics in exam questions.

IV. APPLICATION: WHAT MAKES THE LEARNING OF ALGORITHMS COMPLICATED

The diagnosis of the difficulties experienced by students (or generally, learners) is critical to help them to be successful. Frequently, such difficulties come from the inherent complexity of the concepts or topics being studied. Other difficulties are derived from the lack of adequate background or may come from the individual characteristics of students (lack of concentration, adequateness of mental processes, etc.) [10].

Previous works have acknowledged how important is to take into account the difficulty of concepts and topics for learning effectively [9]. It may help teachers to focus their effort in those concepts that are more difficult for students by, for instance, designing new material, providing exercises for teaching such concepts or motivating student participation through collective discussions on such topics. When evaluating the difficulty of concepts in a learning environment, different approaches have been employed [10]. One is based on the position of the different concepts in a domain ontology, the assumption being that the hierarchical relationship among domain-specific concepts gives an indication of their difficulty. Oliver *et al.* [20], for instance, compare the cognitive difficulty level of computer science courses using the Bloom taxonomy. Another one is based on the frequency of concept-related or domain words in domain-specific texts. The hypothesis here is that the less the word occurs in a language, the higher the complexity of that word will be [5].

Some works have focused on analysing the difficulty of the issues perceived by students and/or teachers and on studying the agreement between the different measures. Boughoula *et al.* [6] use raw clickstream data from video watching sessions of a Coursera MOOC about text retrieval and search engines to discover the difficult topics. Their hypothesis is that the more a video segment is watched, the more difficult it is. Conejo *et al.* [13] performed several experiments to study whether human expert (teacher/student) estimations are similar to difficulty values provided by data-driven techniques. They also analysed the alignment of teachers and students' viewpoints about task difficulty. They used the SIETTE system [14] for their experiments. Their conclusion is that human-based estimations of difficulty are not consistent with those obtained through data-driven techniques. They also found some evidence indicating that students' estimations are better than teachers' ones. Previous works also demonstrate that the perceived level of difficulty of a topic influences the behaviour of students, including the time devoted to study and their motivation [7]. Specifically, the amount of effort expended in performing a task is predicted to increase proportionally with the level of perceived difficulty [15], [26]. As a result, topics or tasks that are perceived as easy will receive little attention and therefore will lead to poor results.

We study the topics and parts of them that are more difficult based on the results of the student assessments. This study includes a detailed analysis of the statistical significance of the results. Then we try to find patterns among the most difficult and the easiest topics, which give us clues about the reasons for the differences in difficulty. The final objective is to facilitate students the learning of the field: as already mentioned previous studies have demonstrated that perceived difficulty by students does not correlate with empiric evidence, so that the grades obtained by students are not consistent with the perceived difficulty of contents. Our study also corroborates this finding. As a consequence, students may not be directing their efforts towards the most difficult contents. Our research aims to estimate the difficulty of contents based on the data, so that the effort and resources can be concentrated in the most difficult contents, without the effect of subjective appreciations. Moreover, since the courses related to data structures and algorithms are usually taught in the first years of the Computer Science programs,

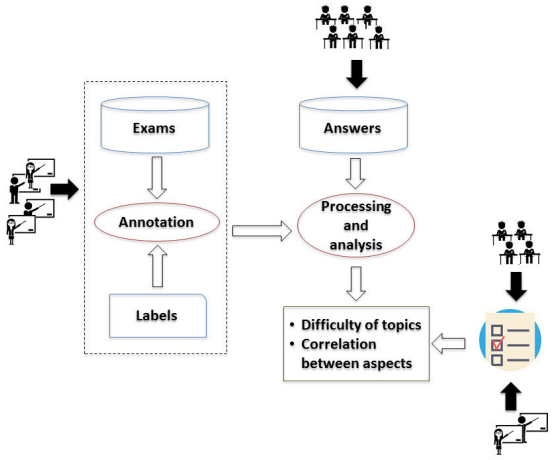


FIGURE 7. Scheme of the data extraction process.

improvements introduced to them have a great impact on the perception and motivation of students and can be very important to reduce student drop out, which is a cause of concern for universities and colleges.

Figure 7 shows a scheme of the methodology followed to perform the proposed analysis. A fundamental part is the preparation of the set of tests from which to extract relationships and statistical data. This step requires, first of all, establishing the set of descriptors or labels that will be assigned to the test questions to characterize them.

Once the labels for the study have been defined, the questions of the exams collected in previous courses have to be labeled, so that each question in the test is associated with a set of labels to which it relates. The system proposed in Section II-B allows to perform this step in an automatic way for future data. The set of questions, labels and the student results for each of them is a valuable resource resulting from this work.

The next step is to analyse the collected data to obtain the success rates for the different topics and aspects considered, the degree of confidence of such results, and the correlations between them.

The first question that we have addressed in this work is whether there are substantial differences in the difficulty of the different topics related to algorithms. Or whether, on the other hand, the differences that can be seen in the results for a topic are arbitrary and change from call to call, depending on particular factors, and therefore the aggregate results, indicate similar levels of difficulty in all the topics and aspects.

Tables 7, 8, and 9 show the percentages of successes, failures and unanswered (blank) questions for the different labels considered. The data are sorted by the success rate. We can see that there are important differences between some of the labels. For example, the topic of the data structure *heap* gets a 81% hit rate, whereas the *quicksort* algorithm of the *divide and conquer* scheme, gets a hit rate of 43%.

Comparing, in Table 7, the results for the *practical* and *theoretical* types of questions, we can see that the rate of success in *practical* questions, 69%, is much higher than the

TABLE 7. Results (right (R), wrong (W) and blank (B) answers) for the general labels, sorted by percentage of right answers.

TOPIC	R (%)	W (%)	B (%)
PRACTICAL	69	21	9
DATA STRUCTURES	69	19	11
SCHEMES	59	32	7
THEORICAL	58	31	9
COST	51	33	15

TABLE 8. Results (right (R), wrong (W) and blank (B) answers) for labels related to algorithmic schemes, sorted by percentage of right answers.

TOPIC	R (%)	W (%)	B (%)
TASK SCHEDULING (GREEDY)	88	8	3
DIJKSTRA	77	19	3
PRIM	71	23	4
GREEDY	70	25	4
KRUSKAL	68	28	3
BACKTRACKING	64	28	6
BRANCH & BOUND	58	36	4
DINAMIC PROGRAMMING	54	25	20
DIVIDE & CONQUER	47	38	14
QUICKSORT	43	45	10

TABLE 9. Results (right (R), wrong (W) and blank (B) answers) for labels related to data structures, sorted by percentage of right answers.

TOPIC	R (%)	W (%)	B (%)
HEAPS	81	16	1
GRAPHS	66	21	12
HASH	60	18	21

rate of success in *theoretical* questions, 58%. We can also observe that students get better results for questions related to data structures (69 %) than for algorithmic schemes (59 %). Results also indicate that questions related to algorithmic cost tend to be difficult (51%).

Regarding the different algorithmic schemes, data indicate that the *greedy* scheme is usually the easiest one (70%), whereas the *divide and conquer* scheme is the most difficult one (47%). It is necessary to keep in mind that the *greedy* scheme does not usually ask for a proof of validity and, in many cases, it is asked in practical questions, such as the application of *Prim*, *Kruskal* or *Dijkstra* algorithms to a specific case.

Regarding data structures, *heaps* seem to be much more affordable, with a 81% success rate, than *hash tables*, with 60%.

The results of this analysis are interesting, but we have to take into account that the amount of data available is different for different labels. There are some topics that have appeared much more often than others in the tests, and thus their results are more reliable. The analysis of the statistical significance of the results is an important point to study, what we tackle in the subsection below.

A. STATISTICAL SIGNIFICANCE

The significance of the results depends on each label and it is important to analyse the confidence for each label. For this

study we have resorted to the *beta* probability distribution:

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}$$

where α is the number of pass marks plus 1, β the number of failing marks plus 1, and x the probability of passing. This distribution allows to obtain the range of values of each data for a given confidence interval. Specifically, it provides the distribution of parameter x, that is, the probability of passing for each label, in our case, assuming that the process by which the data have been generated is binomial.

Figures 8, 9 and 10 show the success rate for the different labels along with its 95% confidence interval.

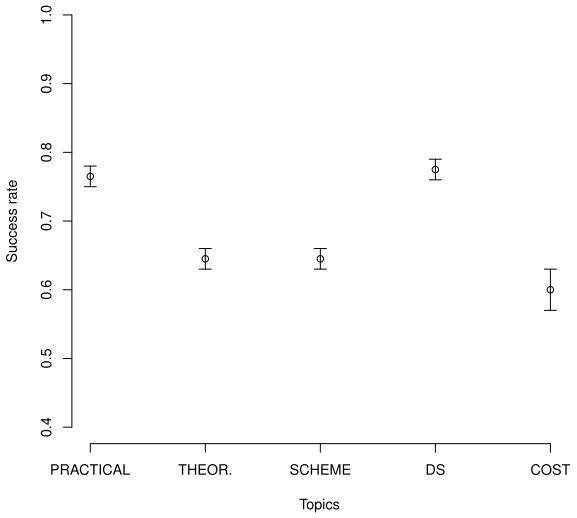


FIGURE 8. Confidence intervals of the results for general labels. THEOR. stands for theoretical questions, DS for data structures, and COST for algorithm cost.

The graph in Figure 8 shows the success rates with its 95% confidence interval for the more general labels, such as those that indicate whether the question is related to data structures (DS) or to algorithmic schemes (SCHEME), whether it is related to PRACTICAL or THEORETICAL concepts, or if it refers to algorithmic COST. In this graph we can see that the range of values of the distribution is small, i.e. it has little uncertainty. It is only a bit broader for algorithmic COST questions, for which the number of questions collected is smaller.

Similarly, Figure 9 compares the labels related to the different algorithmic schemes. In this case, most labels take values in narrow ranges too, although a little wider than those of the general topics, as expected, since we now have fewer data. The range is specially narrow for questions related to the greedy scheme, which are more common (they include Prim, Kruskal, etc.).

Finally, Figure 10 compares the confidence range for the success rate of topics related to data structures. The confidence interval for the topics, *hash tables*, *heap* and *graphs* are very narrow.

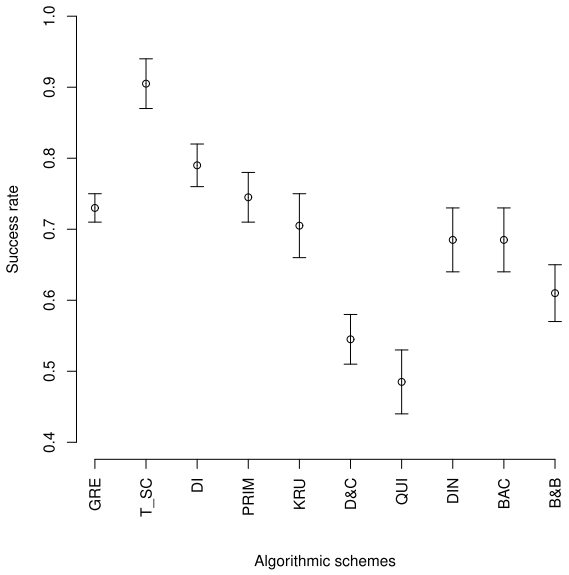


FIGURE 9. Confidence intervals of the results related to algorithm schemes. GRE stands for *greedy*, T_SC for *task scheduling*, DI for *Dijkstra*, KRU for *Kruskal*, D&C for *divide and conquer*, QUI for *quicksort*, DIN for *dynamic programming*, BAC for *backtracking*, and B&B for *branch and bound*.

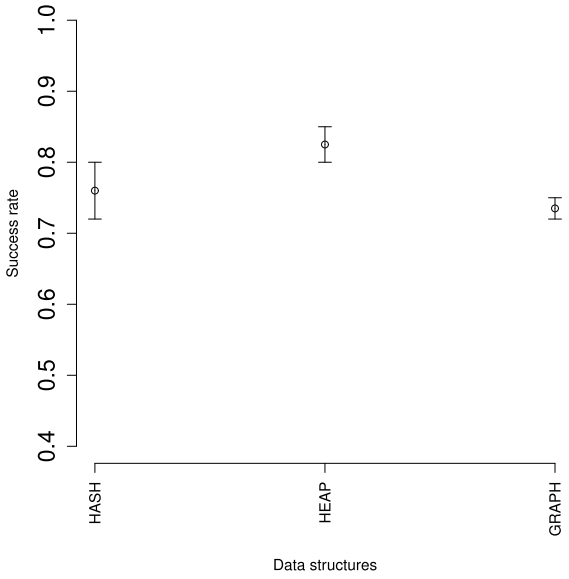


FIGURE 10. Confidence intervals for results related to data structures.

B. CORRELATIONS WITH THE PRACTICAL OR THEORETICAL NATURE OF THE QUESTIONS

Once the analysis has provided a positive answer for the question about substantial differences in the difficulty of the different topics related to algorithms, our objective has been to get clues about the possible reasons for the differences in the levels of difficulty. An interesting question is the influence on the results of the practical or theoretical nature of the questions. In order to quantify the degree of relationship between both variables, the nature of the question and the results, we compute the distance between the measured data

and the expected data if these variables were independent. For that we apply the Pearson Chi-square test:

$$\chi^2 = \sum \frac{(fo_i - fe_i)^2}{fe_i}$$

where *fo* is the observed frequency and *fe* is the expected frequency. In order to apply it, we have built a contingency table for each topic for which there are data on both, practical and theoretical questions.

TABLE 10. Results (right (R), wrong (W), and blank (B) answers) obtained for different topics of the course depending on the practical (PRAC.) or theoretical (THEO.) nature of the questions. The last column shows the significance of the correlation (SC) of that nature with the results. BACK stands for *backtracking*, B&B for *branch and bound*, DS for data structures, D&C *divide and conquer*, and DP for *dynamic programming*. N/A stands for not applicable.

LABEL	TYPE	R	W	B	SC
BACK	THEO.	325(0.64)	145	32	N/A
B&B	THEO.	369(0.58)	231	31	N/A
GREEDY	PRAC.	1020(0.76)	262	49	<8.1e-15
GREEDY	THEO.	539(0.60)	332	52	
DS	PRAC.	1031(0.72)	238	153	0.025
DS	THEO.	876(0.65)	265	166	
HEAPS	PRAC.	573(0.86)	84	9	0.57
HEAPS	THEO.	83(0.86)	10	3	
GRAPHS	PRAC.	350(0.59)	140	102	1.3e-05
GRAPHS	THEO.	518(0.70)	153	66	
SCHEME	PRAC.	1488(0.67)	527	201	< 2.2e-16
SCHEME	THEO.	3359(0.57)	2022	431	
D&C	PRAC.	153(0.49)	120	36	0.46
D&C	THEO.	398(0.46)	329	131	
HASH	PRAC.	108(0.65)	14	42	0.002
HASH	THEO.	275(0.58)	102	97	
DP	PRAC.	315(0.54)	145	116	N/A

Table 10 shows the data for some of the most frequent topics in the tests separated by the practical or theoretical character of the questions. The last column of table 10 shows the significance of the correlation.

In the table we can observe that for several topics, there is only one kind of question, sometimes theoretical, such as *backtracking*, and *branch and bound*; sometimes practical, such as *dynamic programming*. An elementary calculation of the success rate according to the nature of the question indicates whether for a specific topic the students have obtained better results for practical or theoretical questions. Next to the number of hits, in brackets, appears the success rate of each kind of question. It may be observed that practical questions tend clearly to be easier than theoretical ones, being the topic of *graphs* the only exception. In all cases where it has been possible to compute the significance of the correlation, the results are highly significant, except for the data structure *heap*, and the algorithmic scheme of *divide and conquer*. In the case of *heaps*, we can observe that the rate of success of the practical and theoretical questions are exactly the same. In the case of *divide and conquer*, the rates of success for practical and theoretical questions are also very similar.

C. SUBJECTIVE ESTIMATION OF TOPIC DIFFICULTY BY TEACHERS AND STUDENTS

Finally, we have studied whether the perception of students and teachers is a good indicator of the difficulty of the topics, that is, their degree of coincidence with the data. 49 students and 23 teachers have answered a questionnaire about aspects related to the difficulty of topics related to algorithms and advances data structures (see appendix A). The first aspect evaluated both, statistically and heuristically, has been the difficulty of the questions according to their theoretical or practical character. Figure 11 shows the results. It is observed that, for both students and teachers, questions of a practical nature are considerably more difficult than theoretical ones. However, the statistical analysis of the evaluation data leads to a different conclusion: students perform worse on theoretical questions than on practical ones.

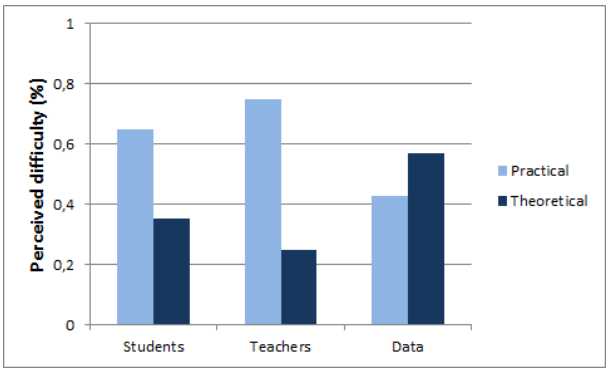


FIGURE 11. Difficulty of topics perceived by students and teachers according to their theoretical or practical character.

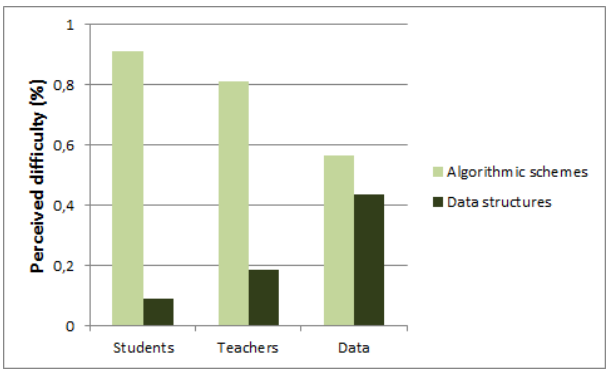


FIGURE 12. Difficulty perceived by students and teachers of data structures versus algorithmic schemes.

In contrast, Figure 12 shows that the perception of the students and teachers about the relative difficulty of the two generic parts of the field (data structures and algorithmic schemes) does coincide with the results produced by the data: the algorithmic schemes involve greater difficulty than data structures. However, it is significant that both, students and teachers, consider the former to be much more difficult than

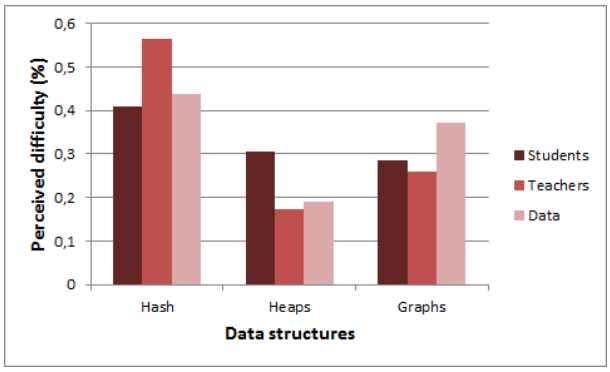


FIGURE 13. Difficulty perceived by students and teachers of the different data structures.

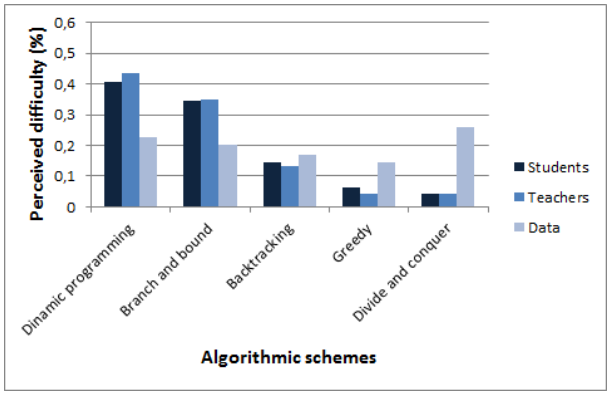


FIGURE 14. Difficulty perceived by students and teachers of the different algorithmic schemes.

the latter, when the results of the students in the tests are only slightly better in the case of data structures.

If we ask about the difficulty of the different topics or concepts that make up each of the two previous blocks of knowledge (see Figures 13 and 14), we observe, once again, that the empirical evidence does not always coincide with the subjective perception of students and teachers. In the case of data structures (graphs, heaps and hash tables), the results of the evaluation tests corroborate the subjective perception, with the *hash tables* being the most complex structure, followed by the *graphs* and, finally, by *heaps*. Hash tables are perceived as significantly more complex than the other two structures, and in fact the scores obtained on hash table questions are significantly worse than those on graphs and heaps. However, in the case of the algorithmic schemes we observe that the least difficult scheme, according to the perception of both students and teachers, is *divide and conquer*, which, however, is the most difficult scheme according to the data. For the rest of the schemes the relative perception of their difficulty coincides with the empirical estimation.

As a conclusion, the evidence based on the data does not always coincide with the perception of students and teachers. Previous work indicates that the perceived level of difficulty of an issue influences student behaviour, including time spent studying and his motivation [7]. Specifically, the amount of effort invested in performing a task will increase proportionally with the perceived level of difficulty [15]. As a result, topics or tasks that are perceived as easy will receive little attention and therefore lead to poor results in the evaluation.

Taking into account these results, we can conclude that it is important to know the difficulty of the topics based on empirical data, in such a way that both students and teachers can direct their efforts towards these elements, correcting possible deviations derived from individual subjective assessments of the difficulty. And to be able to do this analysis it is necessary to have a tool like the one proposed in this work, which simplifies the preparation of the data.

V. DISCUSSION AND CONCLUSIONS

In this work we have presented a system that automatically identify the topics, from a predefined list of educational topics, that are addressed in a text. The system is based on deep learning (DL) techniques and uses different kinds of embeddings, which capture the semantic of the texts. The system has obtained very high results in our experiments (around 90% accuracy), which means that can be reliably applied to educational tasks requiring a process of topic labeling, such as the study of the difficulty of the topics of a subject. The analyses carried out indicate that the best configuration of the input to the model depends on the specificity of the topic to be classified. In general, the neural network that only has the word embeddings as input provides high quality results for many of the topics. For more general topics or aspects, character embeddings can be helpful if they have been trained with enough data. The results are expected to improve as the amount of data increases. It is worth mentioning that with hardly any design effort, just by using pre-trained word embeddings, the proposed system is able to achieve high quality results. Other classical machine learning systems could also achieve these results by using a set of suitable features, but this would require an effort in selecting the features and computing each of them.

Regarding the application of the system to the study of the difficulty of the topics taught in a subject, results have shown that there are indeed topics that are consistently more difficult for most students. Therefore, it is important to analyze the reasons that may be the cause of the greater difficulty of certain topics.

Firstly, it has been observed that algorithmic schemes tend to be more difficult than data structures. We believe there are several reasons for this. On the one hand they require a higher level of abstraction. In many cases, the most appropriate data structure for the problem is known and the challenge is to apply it properly to particular data. However, in the case of algorithmic schemes, the first challenge is the selection of the most appropriate scheme for the problem, which is generally a complex issue, as the results indicate. In addition, the application of algorithmic schemes is more creative and does not follow a particular procedure. It is necessary to choose the best form of representation, the way to reduce the

problem to other simpler ones in the case of the divide and conquer scheme, the bounds in the case of branch and bound, the equations of recurrence in dynamic programming, etc.

With regard to the practical and theoretical aspects, the results clearly show a trend for practical questions to be less difficult than theoretical ones. However, this fact is contrary to the perception of students and teachers. In general, practical questions imply the application of some algorithm or the operation of some data structure to specific data, i.e. to follow a specific procedure. In contrast, theoretical questions require a greater conceptualization, such as in the case of deciding which algorithmic scheme is most appropriate to solve a particular problem. In addition, we consider that the methodological approach of the subject, which involves multiple practical exercises for each data structure and for each scheme presented, also contributes to this trend.

With respect to data structures, results indicate that the simplest are *heaps*, followed by *graphs*, and finally *hash tables* are the most difficult. On the one hand, applying collision resolution methods in *hash tables* to particular data involves procedures with many details that make them difficult and prone to mistakes. On the other hand, *hash tables* are not used to implement the algorithmic schemes, as it happens with *graphs* and *heaps*, so in view of the results, we consider that it is advisable to increase the practical exercises for this structure. The results of the questionnaire for data structures coincide with the results of the evaluation tests. This indicates that students and teachers are aware of their difficulty and can devote the necessary attention to them.

In the case of algorithmic schemes, the results of the students do not coincide with their perception or with that of the teachers. Both, students and teachers, consider *divide and conquer* to be the easiest scheme, whereas data indicate that it is indeed the most challenging one. This discrepancy requires a more detailed analysis of aspects that may affect the difficulty of a topic. For the rest of the schemes, the subjective perception of both, teachers and students, coincides with the data. Another interesting data that has emerged is that the algorithm that implies greater difficulty is the sorting algorithm *quicksort*. This algorithm involves a process with many details that can lead to errors in its application when it is applied manually to a specific case.

We consider the *Divide and Conquer* scheme is perceived as a simpler scheme because its main underlying idea is highly simple: just split the problem into simpler ones. However, the correct application of this idea to particular problems can be very tricky. For example, quicksort algorithm requires applying a complex procedure to combine the solutions provided for each subproblem. In fact, according to the data, quicksort is the most difficult algorithm. For this reason, we believe that it is essential to detect the wrong perceptions of the difficulty of the topics in order to improve the teaching method. This can only be achieved with a detailed comparison of the students' results and the perceptions of students and teachers, an using tools for the automatic annotation of the data, such as the one proposed here.

In the future, we intend to extend the analysis to other knowledge areas and disciplines, and to check their generality and the adaptation requirements. We also intend to continue collecting data on the subject considered in this work and to analyze whether the improvement of resources dedicated to certain subjects improves the results.

APPENDIX A
QUESTIONNAIRE

The questionnaire includes de following questions:

1. Of the following subjects, mark those that you have passed: (Object Oriented Programming | Programming Strategies and Data Structures | Both | None)
2. Have you previously taken an exam on this subject? (No, it's the first time I take this course | No, I took this course before, but never took the exam | Yes, I took this course before and took the exam)
3. In what stage of completion are your practical projects for this subject? (Completed, from last year | Near completed | Just started | Not started yet)
4. Have you seen the video tutorials for the subject? (All of them | Some of them | None of them)
5. Which part of the subject is more difficult for you? (Data Structures | Algorithmic Schemes | Both)
6. Which Data Structure is more difficult for you? (Graphs | Heaps | Hash tables)
7. Which Algorithmic Scheme is more complicated for you? (Greedy | Divide & Conquer | Dynamic Programming | Backtracking | Branch & Bound)
8. Which kind of questions are more difficult for you on exams? (Practical exercises | Theoretical exercises | Both)
9. Do you feel you lack some previous knowledge on those aspects? (you can select more than one): (Computational cost | Recursion | Programming skills | Maths (matrix management, etc), logic or similar)

ACKNOWLEDGMENT

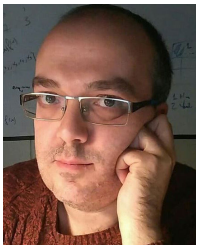
This work has been partially supported by the UNED 2019 project for Innovation Group INEDA (GID2017-1) for the collection of data. The authors thank to Hermenegildo Fabregat for providing the char pretrained embeddings.

REFERENCES

[1] A. S. Alblawi and A. A. Alhamed, "Big data and learning analytics in higher education: Demystifying variety, acquisition, storage, NLP and analytics," in *Proc. IEEE Conf. Big Data Anal. (ICBDA)*, Nov. 2017, pp. 124–129.
[2] H. Aldowah, H. Al-Samarraie, and W. M. Fauzy, "Educational data mining and learning analytics for 21st century higher education: A review and synthesis," *Telematics Informat.*, vol. 37, pp. 13–49, Apr. 2019.
[3] C. Alonso-Fernández, A. Calvo-Morata, M. Freire, I. Martínez-Ortiz, and B. Fernández-Manjón, "Applications of data science to game learning analytics data: A systematic literature review," *Comput. Educ.*, vol. 141, Nov. 2019, Art. no. 103612.
[4] K. Z. Aung and N. N. Myo, "Sentiment analysis of students' comment using lexicon based approach," in *Proc. IEEE/ACIS 16th Int. Conf. Comput. Inf. Sci. (ICIS)*, May 2017, pp. 149–154.
[5] A. Borst, A. Gaudinat, C. Boyer, and N. Grabar, "Lexically based distinction of readability levels of health documents," in *Proc. Med. Informat. Eur. Conf. (MIE)*, Goteborg, Sweden, 2008, pp. 72–75.

[6] A. Boughoula, C. Geigle, and C. Zhai, "A probabilistic approach for discovering difficult course topics using clickstream data," in *Proc. 4th ACM Conf. Learn. @ Scale*, Chicago, IL, USA, 2017, pp. 303–306.
[7] R. L. Capa, M. Audiffren, and S. Ragot, "The interactive effect of achievement motivation and task difficulty on mental effort," *Int. J. Psychophysiol.*, vol. 70, no. 2, pp. 144–150, Nov. 2008.
[8] C. Cardellino. (Mar. 2016). *Spanish Billion Words Corpus and Embeddings*. [Online]. Available: https://crscardellino.github.io/sbwce/
[9] A. Çimer, "What makes biology learning difficult and effective: Students' views," *Educ. Res. Rev.*, vol. 7, no. 3, pp. 61–71, Jan. 2012.
[10] T. H. Chang, Y. T. Sung, and Y. T. Lee, "Evaluating the difficulty of concepts on domain knowledge using latent semantic analysis," in *Proc. Int. Conf. Asian Lang. Process.*, Urumqi, China, Aug. 2013, pp. 193–196.
[11] K. Colchester, H. Hagras, D. Alghazzawi, and G. Aldabbagh, "A survey of artificial intelligence techniques employed for adaptive educational systems within E-learning platforms," *J. Artif. Intell. Soft Comput. Res.*, vol. 7, no. 1, pp. 47–64, Jan. 2017.
[12] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12 pp. 2493–2537, Aug. 2011.
[13] R. Conejo, E. Guzmán, J.-L. Perez-de-la-Cruz, and B. Barros, "An empirical study on the quantitative notion of task difficulty," *Expert Syst. Appl.*, vol. 41, no. 2, pp. 594–606, Feb. 2014.
[14] R. Conejo, E. Guzmán, E. Millán, M. Trella, J. L. Pérez-de-la-Cruz, and A. Ríos, "SIETTE: A Web-based tool for adaptive testing," *Int. J. Artif. Intell. Educ.*, vol. 14, no. 1, pp. 29–61, 2004.
[15] G. H. E. Gendolla and R. A. Wright, "Motivation in social settings studies of effort-related cardiovascular arousal," in *Social Motivation: Conscious and Unconscious Processes*. New York, NY, USA: Cambridge Univ. Press, 2008, pp. 71–90.
[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[17] S. Joksimović, V. Kovanović, and S. Dawson, "The journey of learning analytics," *HERDSA Rev. Higher Educ.*, vol. 6, pp. 37–63, Jul. 2019.
[18] Y. Kim, Y. Jernite, D. A. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, Feb. 2016, pp. 2741–2749.
[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. 1st Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds., Scottsdale, AZ, USA, May 2013, pp. 1–12.
[20] D. Oliver, T. Dobele, M. Greber, and T. Roberts, "This course has a Bloom rating of 3.9," in *Proc. 6th Australas. Conf. Comput. Educ. (ACE)*, Darlinghurst, NSW, Australia, vol. 30, 2004, pp. 227–231.
[21] A. Ortigosa, J. M. Martín, and R. M. Carro, "Sentiment analysis in facebook and its application to e-learning," *Comput. Hum. Behav.*, vol. 31, pp. 527–541, Feb. 2014.
[22] C. Romero and S. Ventura, "Educational data mining: A survey from 1995 to 2005," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 135–146, Jul. 2007.
[23] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," *WIREs Data Mining Knowl. Discovery*, vol. 10, no. 3, p. e1355, May 2020.
[24] C. P. Rosé and O. Ferschke, "Technology support for discussion based learning: From computer supported collaborative learning to the future of massive open online courses," *Int. J. Artif. Intell. Educ.*, vol. 26, no. 2, pp. 660–678, Mar. 2016.
[25] S. Van Goidsenhoven, D. Bogdanova, G. Deeva, S. V. Broucke, J. De Weerdt, and M. Snoeck, "Predicting student success in a blended learning environment," in *Proc. 10th Int. Conf. Learn. Anal. Knowl. (LAK)*, New York, NY, USA, 2020, pp. 17–25.
[26] R. A. Wright and L. D. Kirby, "Effort determination of cardiovascular response: An integrative analysis with applications in social psychology," *Adv. Exp. Social Psychol.*, vol. 33, pp. 255–307, Jan. 2001.
[27] X. Zhang, Y. Meng, P. O. de Pablos, and Y. Sun, "Learning analytics in collaborative learning supported by slack: From the perspective of engagement," *Comput. Hum. Behav.*, vol. 92, pp. 625–633, Mar. 2019.

LOURDES ARAUJO is a Full Professor with the Universidad Nacional de Educación a Distancia (UNED) and the Head of the Department of Lenguajes and Sistemas Informaticos. She belongs to the Natural Language Processing and Information Retrieval Group. She has taken part, sometimes as a leader, in several research projects related to applications of natural language processing to different domains. She has authored more than 100 research articles. Her current research interests include natural language processing and information retrieval and their application to the fields of education and medicine.



FERNANDO LÓPEZ-OSTENERO received the B.S. degree in mathematics in 1997 and the Ph.D. degree in industrial engineering in 2002. He has been a Senior Lecturer with the Department of Languages and Computer Systems, Universidad Nacional de Educación a Distancia (UNED), since 1998, where he is a Researcher with the UNED IR and NLP Group, and has participated in several research projects within this group. His main research interests include natural language processing, information retrieval, and educational applications.

JUAN MARTÍNEZ-ROMO received the Ph.D. degree in computer science from the Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain. He is an Associate Professor with the Departamento de Lenguajes y Sistemas Informáticos, UNED, since 2007, where he is a part of the Natural Language Processing Group. He currently holds the position of Academic Secretary of the ETSI Informática, UNED. He has participated in nine research projects funded in public calls, six projects with private entities, and has been reviewer of different indexed journals and congresses in the area. His main interests include natural language processing, information retrieval with adversary, bioinformatics, educational technologies, spam detection, web search, e-learning, and use of mobile devices.

LAURA PLAZA is a Senior Lecturer with the Universidad Nacional de Educación a Distancia (UNED) and a Researcher with the UNED IR and NLP Group. Her research interests include different fields of NLP, including summarization, information retrieval, and sentiment analysis, with especial interest in educational applications. She has published in more than 50 international journals and conferences, and has participated in different funded projects and worked in several international companies.