Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Teaching activity recognition in lectures delivered at UPV

Degree final work

Degree in Computer Engineering

Author: Alberto Romero Fernández

Tutor: Eva Onaindía de la Rivaherrera

Course 2019-2020

Acknowledgements

I would like to express my deepest appreciation for Dr. Eva Onaindia for proposing this topic to me as well for her guidance along the entire development of it.

I would like to extend my sincere thanks to Daniel Diosdado who was developing his PhD. project under Dr. Onaindia's supervision and helped in the training of the *XLM-Roberta* model.

In addition, I'd like to acknowledge the effort of Jorge, my brother, for proof reading and suggestions.

Finally, I wish to thank my parents for their love and encouragement; without them I would have never become who I am today.

This work has been partially supported by the project Class Activity Recognition (PROMETEO/2019/111 [1]) and Spanish MINECO project TIN2017-88476-C2-1-R.

Resum

En el present TFG es proposa la construcció d’un sistema per al reconeixement d’activitats docents a partir de transcripcions d’gravacions d’àudio a l’aula. Per a això, identifiquem un conjunt d’activitats docents que cobreixen tots els tipus de discurs acadèmic que un docent pot emprar a l’aula quan imparteix un curs, com per exemple, "teoria", "resolució d’exercicis", "exemples pràctics del món real", ”interacció entre professor i estudiant", "organització i gestió de l’assignatura", etc. A més, creem un *dataset* a partir de les transcripcions de classes gravades amb el servei VideoApuntes de la UPV i etiquetem segments de les transcripcions amb el tipus de discurs corresponent. Posteriorment, dissenyem una tasca de classificació que es resol amb el model XLM-RoBERTa, una versió millorada de BERT (Bidirectional Encoder Representations from Transformers) sobre el que s’ha dissenyat una capa de classificació. Els resultats mostren un alt nivell de precisió en la classificació de segments de text del discurs acadèmic que empren els professors en la impartició d’assignatures. Finalment, es pretén dissenyar un mètode per a identificar el tipus d’activitat docent que es reflecteix en un segment de transcripció a partir d’un arbre de classificació binari.

Paraules clau: Activitats docents, Transcripcions, Classificació, Xarxes Neuronals, Processament de Llenguatge Natural

Resumen

En el presente TFG se propone la construcción de un sistema para el reconocimiento de actividades docentes a partir de transcripciones de grabaciones de audio en el aula. Para ello, identificamos un conjunto de actividades docentes que cubren todos los tipos de discurso académico que un docente puede emplear en el aula cuando imparte un curso tales como por ejemplo, "teoría", ’resolución de ejercicios",”ejemplos prácticos del mundo real", ”interacción entre profesor y estudiante", ”organización y gestión de la asignatura", etc. Además, creamos un *dataset* a partir de las transcripciones de clases grabadas con el servicio VideoApuntes de la UPV y etiquetamos segmentos de las transcripciones con el tipo de discurso correspondiente. Posteriormente, diseñamos una tarea de clasificación que se resuelve con el modelo XLM-RoBERTa, una versión mejorada de BERT (Bidirectional Encoder Representations from Transformers) sobre el que se ha diseñado una capa de clasificación. Los resultados muestran un alto nivel de precisión en la clasificación de segmentos de texto del discurso académico que emplean los profesores en la impartición de asignaturas. Por último, se pretende diseñar un método para identificar el tipo de actividad docente que se refleja en un segmento de transcripción a partir de un arbol de clasificación binario.

Palabras clave: Actividades docentes, Transcripciones, Clasificación, Redes Neuronales, Procesamiento de Lenguaje Natural

Abstract

In this project, we propose to build a system for recognizing teaching activities from automatic transcriptions of classroom video recordings. To this end, we identified various teaching activities that cover the nature of the lecturer discourse when giving a course eg. ′theoretical explanation′, ′problem solving′, ′real-world practical example′, ′interation lecturer-student′, ′course-related asides′, etc. We labeled a dataset of lecture transcriptions from the VideoApuntes repository of UPV and we solved a classification task with the XLM-RoBERTa model, an improved version of BERT (Bidirectional Encoder Representations from Transformers) with a classification layer on top of it. The results will show the high accuracy in classifying text segments of the discourse. Finally, we aim to conduct an experiment in order to identify the type of teaching activity reflected in a text segment using a binary decision tree.

Key words: Teaching activities, Transcriptions, Classification, Neural Networks, Natural Language Processing

Contents

List of Tables

CHAPTER 1
Introduction

1.1 Motivation

University lectures have been taught in the same way for hundreds of years, where professors have a monologue about the topic being exposed and occasionally a student may interrupt for questioning. This dominant trend is changing rapidly with new advances in technologies. Nowadays, students may learn where and when they want and, more importantly, at a pace of their choosing, thanks to lecture recording. This has been especially notable during the ongoing SARS CoV-2 pandemic. Building on this basis, we propose a project to design a tool that will automatically classify the audio transcription of a lecture into the teaching activities employed by the teacher for delivering the lecture and thus improving the experience of students when watching prerecorded lectures.

This project stems from a PROMETEO research project entitled 'CAR: CLASS ACTIVITY RECOGNITION' [1] supported by the Conselleria d'Educació Cultura i Esport of Generalitat Valenciana. The CAR project aims to analyze the degree of engagement of the listeners (students) to a class given by an orator (lecturer). The primary focus is on helping teachers to visualize and improve their discourse management skills. It also puts focus on the students and takes a step towards the elicitation of a student participation model to identify reactive/proactive and female/male behaviors during lectures as well as determining different participation models depending on the type of lecture (theory, exercises, practice). The main objectives of CAR are:

1. To provide high-quality automatic transcriptions of classroom video recordings,
2. To provide an automatic classification of activities from classroom transcriptions,
3. To provide a behavioral pattern model of students and lecturers,
4. To provide an academic performance assessment model.

1.2 Objectives of the thesis

This project is positioned within the second objective of the research initiative CAR mentioned above; i.e., *to provide an automatic classification of activities from classroom transcriptions*.

The starting point of this project is the automatic transcriptions of audio recordings of lectures delivered at UPV and stored in a repository of the University. The objective is to analyze the audio & transcriptions files and recognize the teaching activity the lecturer is employing in class at each time. To this end, the particular sub-objectives of this work are:

- to propose a classification of teaching activities commonly used in university lectures,
- to build a system for recognition of teaching activities through automatic transcriptions of classroom audio recordings,
- to analyze the differences between an automated recognition of teaching activities and an experienced human's approach to the same classification task.

1.3 Structure

This thesis is structured as follows:

- In chapter 2 we give some context on what the state of the art is in terms of spoken academic discourse, giving some information on how it is divided and what the literature says about it. It also describes what transformer-based models are in use nowadays, in particular the BERT (Bidirectional Encoder Representations from Transformers)) Transformer and its family.
- In chapter 3 we describe the problem and define where all the data we used is taken from, how the transcriptions are made, how the data segmentation and labeling is done and describe what each label is defined as and how one can identify them.
- In chapter 4 we dive into the neural network classification, on the specifications of the *XLM-RoBERTa* model, tokenization, experimental evaluation and its results.
- In chapter 5 we develop a human-like classification binary tree based off of the shared vocabularies between classification segments and possible labels. We finally compare the results from this method with the neural network method.
- In chapter 6 we describe the conclusions we extract from the whole project and suggest some adjustments and further improvements for a future work.

CHAPTER 2
State of the Art

The aim of this project is to build a system that recognizes teaching activities from automated lecture transcriptions, particularly from lectures delivered at the Universitat Politècnica de València (UPV). In order to position our work within the relevant literature, we will study works that are principally concerned with two topics: the spoken discourse in academic lectures and the use of neural network models for common Natural Language Processing tasks.

2.1 Spoken academic discourse

In linguistics, the term **genre** refers to types of spoken and written discourse recognized by a discourse community. Examples of spoken and written genres include lectures, conversations, speeches, notices, advertisements, novels, diaries, shopping lists, paper or poster presentations, seminar discussions, research articles, interviews, questions in a lecture, justification of a research proposal, and many more. Since this project is devoted to examining the speech used in classes delivered at UPV, our focus is on the **academic spoken discourse** genre.

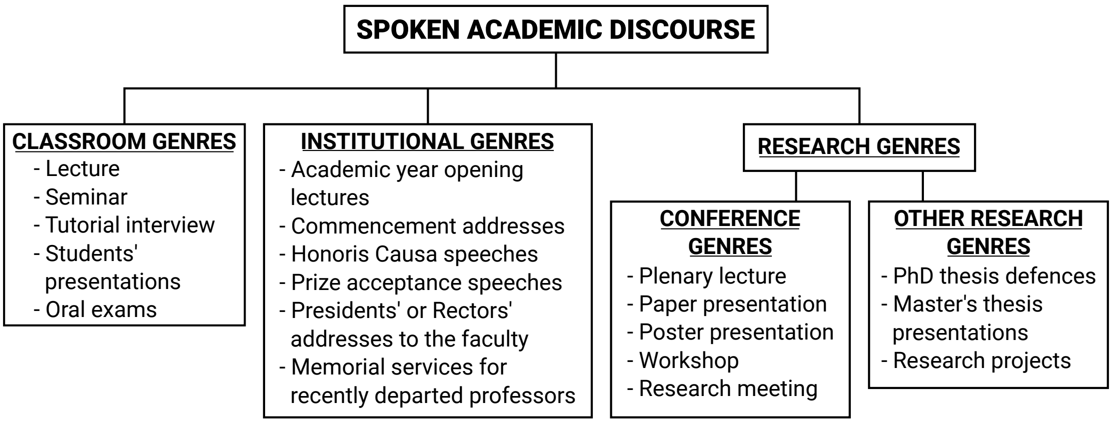


Figure 2.1: Classification of academic genres according to their purpose (taken from [2]).

Figure 2.1 shows a classification of spoken academic genres according to criteria of purpose [2]. This classification identifies (1) research genres, as faculty's academic life involves presenting papers at conferences, defending doctoral thesis or even research and lecturing in other universities; (2) institutional genres, mostly used by university representatives and authorities in official and institutional speeches; and (3) **classroom genres**, which are regarded as paramount for both students and faculty. Hence, the language and speech used in a PhD. thesis defence differs from the ones used, for instance, in an academic year opening speech and both in turn differ from those used in classroom genres.

Among the classroom genres, the seminar, tutorial, presentation and oral exams are one kind of interactive genres as they involve a higher level of interaction between the presenter and the audience which the activity is addressed to [2]. In contrast, the academic lecture is mostly considered an expository genre even though interaction and communication between teachers and students can take place as well.

In this work, we put our attention specifically on the **academic lecture** genre among all possible classroom genres.

2.1.1. Academic lecture

Among the classroom genres, the **lecture** is regarded as a central spoken genre in higher education in Europe and many countries worldwide. Research on the discourse of lectures is becoming more and more relevant due to the increasing internationalization of higher education both from the point of view of students and lecturers [3].

The discourse community of our study is the university faculty and students who attend lectures. Students attending a lecture need to listen and understand first to be able to take notes. Additionally, while lectures mainly belong to the expository genre and have a monological style, there are parts of the lecture which involve an dialogue with the students, for example in the form of question-answer interaction initiated by a student or by the lecturer.

Academic lectures can be considered as an oral/literate mixture, where oral refers to stereotypical speaking such as conversation, and literate refers to stereotypical writing as in academic prose. That is, lectures share several situational characteristics of both academic prose and face-to-face conversations [4]. Lectures have a highly informational focus, similarly to academic prose, and, at the same time, have interactive features as they are delivered under on-line production circumstances that resemble face-to-face conversations in the spoken mode [5]. And it is precisely this twofold aspect of the lectures that we are particularly interested in analyzing.

In order to categorize a lecture, we need to define features that capture the informational purpose of the lecture as well as displaying features of the spoken discourse. In this line, some researchers have examined the macro-structure of university lectures and the micro-features that contribute to this structure [6]. According to Young, the phases which mark university lectures are [6]:

- **Interaction**: this is an important feature that indicates to which extent lecturers maintain contact with their audience in order to both reduce the distance between themselves and their listeners, and to ensure that what has been taught is in fact understood.
- **Theory or Content**: this is used to reflect the lecturer's purpose, which is to transmit theoretical information.
- **Examples**: it is in this phase in which the speakers illustrate theoretical concepts through concrete examples familiar to students in the audience.

In more recent research, the academic lecture is characterized, among others, by the following features [7]:

1. The purpose of a lecture is to convey knowledge to a large number of students.
2. Lecturers are able to give **examples of practical application** and to **relate personal experience** to the content of the lectures.
3. An appropriate setting for a lecture is one in which the teacher not only presents information to the audience but also **expresses their attitudes and evaluation of materials**.
4. A lecture is a mixing of genres: formal and informal language, spoken and written (text in slideshows or other forms of text).
5. A lecture is determined by the individual stylistic features of the presenter; e.g. preferring reading aloud, whereas another speaker choosing to **interact with students** and to engage them in communication.
6. There is no direct distinction between a sentence and an utterance in the speech. This may cause a problem with grabbing the knowledge while listening to an academic lecture if pauses are not properly used during the discourse.

With all this information in mind, we will come up with a set of features that help us identify the part of the speech of a university lecture devoted to convey informational knowledge (theory and examples), and the part of the speech devoted to informal registers (interaction).

2.2 Transformer-based models for NLP tasks

Natural Language Processing (NLP) is a research field which studies the interactions between computers and natural human language like speech or text, with the aim of building an agent capable of understanding and/or producing human language in a human-like way.

NLP techniques are used in multiple applications of textual data analysis such as in social media to recognize fake news, offensive language or sentiment analysis as well as in tasks of classification and organization of collections of documents for topic analysis. Traditional NLP tasks for the treatment of textual data include tasks like morpho-syntactic tokenization, lemmatization and stemming, stopword removal, syntactic parsing, part-of-speech tagging, semantic labels, word sense disambiguation, etc. The application of these linguistic-based methods requires to have prior knowledge of the language.

Deep Learning (DL) methods have proven to outperform traditional linguistic-based methods by replacing them by end-to-end architectures where no prior knowledge on the language is needed. In contrast, DL methods have the ability to learn the underlying linguistic nature of the text provided that a large amount of data is available. In the following sections, we briefly summarize the most relevant advances in DL methods applied to NLP.

2.2.1. Transformer models in NLP

Over the past three years, the emergence of transformer-based language models has revolutionized the field of NLP. The **Transformer** is a DL model introduced in 2017 that has proven to be especially effective for common NLP tasks. It is a novel encoder-decoder architecture for sequence-to-sequence (Seq2Seq) models that is based solely on *attention* mechanisms [8].

Seq2Seq models consist of (1) an encoder that takes an input sequence and maps it into a higher dimensional space and (2) a decoder that takes the abstract vector returned by the encoder and turns it into another sequence. The output sequence of the decoder can be a sentence in a different language from the input sequence (translation task), a sequence of symbols, etc. One choice for training the encoder and the decoder of a Seq2Seq model is to use a Long-Short Term Memory (LSTM) neural network for each of them (one for the encoder, one for the decoder). This dominant trend for training Seq2Seq models is based upon complex recurrent or convolutional neural networks in an encoder-decoder configuration [9].

The groundbreaking idea proposed in [8] lies in exploiting **attention mechanisms** which put the attention on the parts of the sequence that are relevant at each step. That is, humans not only read words but their minds hold the important aspects of a sentence to provide context. Capturing the relationships among words in a sentence is vital to understand natural language. And it is precisely here where the Transformer comes into play.

The attention mechanism in DL is based on putting the focus on the relevant aspects, paying more attention to certain factors when processing the data. The attention mechanism is a part of a neural architecture that enables the dynamical highlighting of relevant features of the input data. Specifically, in NLP, these relevant features typically represent a sequence of textual elements [10].

At the time of writing this document, there exist several leading transformer-based language models, namely BERT [11], GPT-2 [12] and TRANSFORMER-XL [13]. We opted for using the BERT model since one of its latest versions has shown to offer very good results as well as an efficient memory saving model.

2.2.2. BERT Transformer and its family

BERT (Bidirectional Encoder Representations from Transformers) is a recent language representation model designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right contexts of a target word in all layers. In consequence, BERT can be easily fine-tuned by adding just one output layer to create state-of-the-art models for a wide range of NLP tasks [11].

BERT's model architecture is composed of multiple layers of bidirectional Transformer encoder blocks. This Transformer blocks are encoder-decoders which employ stacked self-attention and point-wise, fully connected layers for both the encoder and the decoder [14]. The model is pre-trained over two unsupervised tasks: (1) Masked Language Model (MLM), which consists in masking a percentage of the input tokens at random and predict those masked tokens, and (2) Next Sentence Prediction (NSP), in which the input is made of two sentences, A and B, and 50% of the time B is the actual sentence that comes after sentence A.

RoBERTa (Robustly optimized BERT approach) [15] brings a significant improvement over BERT, matching or exceeding the performance of the BERT-based methods. RoBERTa increases the amount of training data and the number of training passes through the data, and it modifies the pre-training regime as follows:

- Dynamic MLM: instead of performing the masking once during preprocessing, the masking is performed every time the model is fed a sequence.

- Full-Sentences without NSP loss: instead of two sentences, the model is fed full contiguous sentences and the NSP Loss is removed.
- Large mini-batches: the batch size is increased from 256 to 8192 samples.
- Larger byte-level Byte Pair Encoding: instead of using a unicode character-level BPE vocabulary of 30k subwords units, RoBERTa uses a larger byte-level BPE vocabulary of 50k subwords units.

The **XLM (Cross-Lingual Language Model)** [16] extends previous efforts to obtain state-of-the-art results on cross-lingual classification of unsupervised and supervised machine translation by processing all languages with the same shared vocabulary created through Byte Pair Encoding and two new training tasks besides MLM:

- Causal Language Modeling: this task consists in training the language model to learn the probability of a word given the previous words in a sentence.
- Translation Language Modeling: similar to Masked Language Modeling, the model is fed two sentences that are direct translations in different languages with randomly masked words.

Finally, **XLM-RoBERTa** is a multilingual model resulting from training a transformer-based MLM on 100 different languages and using more that two terabytes of filtered CommonCrawl data [17]. It does not require to understand which language is used and determines the correct language from the input ids.

XLM-RoBERTa achieved a new state-of-the-art on a variety of cross-lingual benchmarks while also obtaining a comparable performance with monolingual models like RoBERTa. The improvement stems from:

- Increasing the training data by several orders of magnitude from Wiki-100 to a total of 2.5 TB,
- Using a large vocabulary size of 250k subwords and changing the Byte Pair Encoding to SentencePiece with a unigram language model,
- Training the model in the MLM task with 100 languages using the same sampling distribution as XLM but changing learning rate to $\alpha = 0.3$.

The authors of XLM-RoBERTa remark the surprising effectiveness of multilingual models over monolingual models, and show strong improvements of multilingual models on low-resource languages [17].

CHAPTER 3
Problem description and data processing

3.1 Overview of the Problem

The problem we address in this project is about the recognition of academic lecture activities from automatic transcriptions of classroom audio recordings. The process of classifying the academic activities is tackled in four stages:

1. **Selection of audio recordings**. In this step we select the audio recordings of lectures delivered at UPV which are stored in the repository VIDEOAPUNTES. The quality of the audio recordings is a determinant factor in the quality of the corresponding transcriptions. Section 3.2 presents the list of selected audio recordings and section 3.3 outlines the automated transcription process that yields the transcript files.

2. **Segmentation and labeling**. This is the task devoted to dividing the text of the transcriptions into meaningful units (segments) and labeling each segment with one type of academic feature. This task is undertaken by a group of human labelers and it is detailed in sections 3.4 and 3.5.

3. **Classification model**. In this stage, we train a Neural Network (NN)-based model to learn classify the segments of the transcriptions. The NN model builds upon the XLM-RoBERTa multi-lingual model explained in the previous chapter. Chapter 4 offers a thorough explanation on the classification task and the obtained results.

4. **Human-like validation of classification results.** We were also interested in comparing the results obtained with the NN model versus a more human-like classification. To this end, we designed an alternative classification model for only a subset of labels representing particular academic lecture activities, and we performed a comparison of the results obtained with the NN and the ones obtained with the human-like classification.

3.2 Audio recordings from VideoApuntes repository

The Universitat Politècnica de València (UPV) provides an automatic recording system that allows lecturers to record their lectures and upload the videos to the internet for the students to watch them back. VIDEOAPUNTES [18] is the name of the service provided by

UPV to make desktop and/or webcam recordings (Screencast) of a lecture from a computer located at home or from a tablet or mobile phone. The VIDEOAPUNTES service can only be used in rooms equipped with the required facilities for the automatic recording. At the time of writing this document, VIDEOAPUNTES is available in 36 lecture halls of UPV across all three campuses [18].

Lecturers willing to use this service set up and schedule the recordings of their lectures, including video, audio and multimedia of a lecture. VIDEOAPUNTES works with *lavalier* microphones, also referred to as lav mic, a body mic, a clip mic, or a personal mic, that lecturers need to clip onto their clothing. The sound reproduction of this type of microphones is sensitive to voice proximity, therefore a sudden move or head turn by the lecturer may largely affect the audio quality. Furthermore, since the recording of a class is scheduled at a particular time slot, some recordings may have silent parts or recording faults due to, for instance, a rescheduling, a delay in the starting of the class or a recording hardware malfunction. It may also be the case that recordings contain background noise or microphone feedback, especially at the starting and/or end of the lecture. This usually happens if the beginning of the lecture is delayed or when the lecturer is setting up the equipment. Other audio problems that make it hard for transcriptions stem from overlapping conversations between lecturers and students or among the students themselves. These side conversations typically cause inaudible parts and therefore problems for transcriptions.

On a first instance, our idea was to use the students' transcriptions as well as the professors'. However, there exist many situations in which students ask a question or talk during a lecture and the microphones do not pick up their voice and hence this is not reflected in the transcription file. Consequently, the focus of this work revolves around the academic discourse of the lecturer.

Since this project heavily relies on NLP tasks, it is vital for us to have the best possible audio transcriptions from the recordings. Therefore, we are particularly interested in selecting good-quality audio recordings that contain as few issues as possible so as to ensure good transcriptions.

3.2.1. Data collection

The VIDEOAPUNTES repository contains video and audio recordings of multiple lectures of the UPV faculty. UPV mostly offers engineering degrees, which means that a very large number of the courses taught at UPV are about **technical** and **scientific** matters.

Our aim was to select a broad selection of subjects to ensure as much diversity as possible and no bias towards a specific field. We included recordings of lectures that may be qualified as rather **theoretical-based** courses, and others that have a more **hands-on** orientation. We were also interested in gathering recordings that covered **male** and **female** lecturers so as to have a sample that accounts for gender balance. It is important to keep in mind as well that the overall duration of the audio recordings for all the courses must be approximately the same.

Finding a good sample of female professor recordings was challenging due to a general absence of women in STEM subjects. Fortunately, we got to select a balanced number of hours of lecture recordings for male and female lecturers across a broad diversity of subjects.

Table 3.1 shows the seven different subjects covered by the selected video files along with the recording time (in minutes) and the academic degree in which they are delivered. The three most relevant criteria we used to select the recordings are:

- good-quality audio that facilitates transcriptions as accurate as possible,
- subject diversity: many of the subjects are from the Telecommunications degree but they cover a significantly large and diverse corpus of concepts and contain a vast range of vocabulary,
- gender variety so as to avoid a bias towards how gender may affect the way of teaching.

Tables 3.1 and 3.2 depict some statistics of the collection of video files retrieved from VIDEOAPUNTES for our analysis. Table 3.2 shows the seven courses or subjects covered by the transcription files, the total number of video recordings (total number of delivered letures) as well as the total number of lecturers involved in the recordings divided by gender.

Course name	**Male**	**Female**	**Academic Degree**
Statistics	-	345′	Public Administration
Electronic Devices	-	301′	Telecommunications
Mathematics	302′	332′	Telecommunications
Digital Signal Treatment	-	311′	Telecommunications
Oceanographic Physics	302′	-	MSc. Marine Ecosystems
Networks and Teledetection	353′	-	MSc. Marine Ecosystems
Microprocessed Systems	360′	-	Telecommunications
Total	**1499′**	**1483′**	-

Table 3.1: Division of minutes per course and per gender.

Number of Subjects	7
Number of Videos	27
Number of Lecturers	8
Number of Male Lecturers	4
Number of Female Lecturers	4

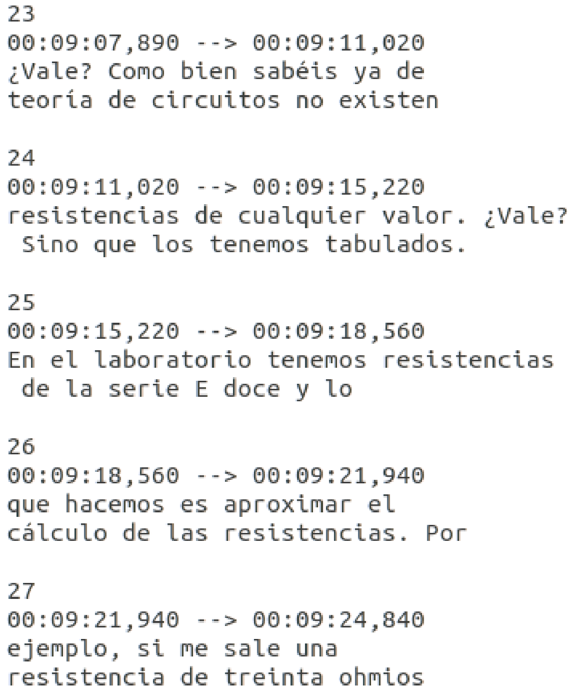
Table 3.2: Analysis of the data from the videos used.

3.3 Transcriptions from audio recordings

As commented in the previous section, we selected 27 recordings of lectures for our purposes. Once selected the audio files, we retrieved the corresponding transcriptions files using an automatic transcription software.

The MLLP transcription and translation platform is an online platform for automated and assisted multilingual media subtitling and text translation created by UPV's Machine Learning and Language Processing (MLLP) research group. It provides support for the transcription and translation of video, audio and the full content of massive open online courses (MOOCs). It also integrates other MLLP-developed technologies such as Text-to-Speech synthesis for enhanced accessibility [19]. For this project 6 out of the 27 total number of videos selected were human-reviewed, these videos have a much more precise and accurate transcription and are very helpful for segmentation and labeling.

An excerpt of the output file returned by the MLLP tool that contains the transcription of one of the recorded lectures is shown in Figure 3.1. In this figure, the transcription contains various sections numbered from 23 to 27. This number represents the order in which the text of each section will appear on screen as a caption, so the text below number 23 is the 23rd caption, 24 is the 24th and so on. The second line represents the time interval in format HH:MM:SS,ms at which the caption is found in the video file. The following lines contain the transcription of the audio (caption) as it will be seen by the viewer on the screen. Currently, the MLLP tool supports 10 transcription languages including, for instance, Catalan, English or Italian. We will only use transcriptions into Spanish since all the recordings are lectures delivered in Spanish.



```
23
00:09:07,890 --> 00:09:11,020
¿Vale? Como bien sabéis ya de
teoría de circuitos no existen

24
00:09:11,020 --> 00:09:15,220
resistencias de cualquier valor. ¿Vale?
 Sino que los tenemos tabulados.

25
00:09:15,220 --> 00:09:18,560
En el laboratorio tenemos resistencias
 de la serie E doce y lo

26
00:09:18,560 --> 00:09:21,940
que hacemos es aproximar el
cálculo de las resistencias. Por

27
00:09:21,940 --> 00:09:24,840
ejemplo, si me sale una
resistencia de treinta ohmios
```

Figure 3.1: Sample output of MLLP processing of transcriptions.

3.3.1. Manual revision of transcriptions

Many transcription services include, besides the automated transcription, a manual review process to ensure accuracy and a high success rate with the transcription. Around 10% of the video transcriptions made with the MLLP tool and stored in VIDEOAPUNTES went through a human review process. Specifically, 6 out of the 27 selected videos were manually reviewed (a more detailed explanation is given in Table 4.2). This means that the transcriptions of these lectures are much more reliable and accurate to the original discourse of the speaker.

The MLLP research group uses a tool called *Transcriber* [20] to help them transcribe the videos. Transcriber is a tool for assisting the manual annotation of speech signals. It enables segmenting, transcribing and labeling of long-duration speech recordings all via a user-friendly graphical user interface. It also provides facilities for annotating speech turns, topic changes and acoustic conditions. Transcriber is more specifically designed for the annotation of broadcast news recordings but its features are found generally useful for the transcription and annotation of speech signals for linguistic research.

The manual revision done by the MLLP members follow some basic transcription guidelines like:

- when possible, each segment must represent a complete meaningful sentence,
- correction of repetitions in the discourse (e.g., "there was a a a problem with ..."),
- correction of hesitations (e.g., "hmmmmm ..."),
- correction of long silences (more than a second) in the middle of a sentence,
- correction of numbers (must be written in words), variable names, equations,
- etc.

3.4 Data segmentation and labeling

In this section, we explain the characteristics we wish to analyze from the academic discourse used by lecturers in the transcribed lectures. In subsection 3.4.1 we present the categories (labels) we propose to classify the discourse of a university lecture. Section 3.4.2 details the protocol we used for segmenting the transcription files and labeling each segment.

3.4.1. Label hierarchy of an audio & transcription file

In section 2.1 we discussed the structure and features that some linguists distinguish in university lectures. Our aim is to define a set of **lecture features** based on language choices that help us identify the contents and purpose of segments of the academic discourse from a transcription file.

For the definition of the lecture features we decided to not only use the transcription files but also listen to the audios as an aid. Hence, we first watched several videos of lectures from YouTube and VideoApuntes, listening to the audio and reading the transcriptions to figure out the nature and purpose of the academic discourse along the transcription. We learned that situations of several kinds may appear as a result of the quality of the audio recordings, the lecturer's speech itself, the students' chat and the content of the speech, all of which, in turn, affect the quality of transcriptions.

After a thorough analysis, we put forward the hierarchy of labels shown in Figure 3.2, where the colored nodes represent the hirearchy of the **lecture features** and the white nodes are labels that help us identify sections of the audio & transcription files. Specifically, the white nodes are labels to denote parts of the audio file which do not have a readable transcription. In the following, we explain the label hierarchy of Figure 3.2.

Level 1: filtering out sounds from the audio file. We found that the audio files of some recordings contained corrupted sections or unwanted sounds due to the lecture starting several minutes after the start of the recording, the recording being suddenly cut off, background noise, errors in the recording or microphone feedback. We identify these damaged sections of the audio file as *Miscellaneous* and the rest is classified as audio that belongs to the *Lecture*.

Level 2: speaker identification. The purpose of the labels at level 2 is to distinguish the parts of the file in which the speaker (the lecturer) is talking from those in which they are not. The four labels, *Indistinct Chat*, *Speaker*, *Pause* and *Multimedia* are easily identified by just listening to the audio. Particularly, these labels are used to mark sections of the audio file that contain an indistinguishable speaker (*Indistinct Chat*, *Multimedia* and *Pause*) and audio sections that clearly identify the target speaker (*Speaker*). In general, we can affirm that, except for the *Speaker*, the rest of labels represent sections of the audio that do not comprise enough speech or information and thus cannot be transcribed.

Level 3: lecture-audience relationship. From level 3 and downwards, all of the features can be retrieved from the transcription file, as after the filtering of labels at level 1 and level 2 we obtain a file that exclusively contains the discourse of the speaker.

The four labels at level 3 denote different ways that the lecturer can address the students, thus creating a different atmosphere depending on the type of discourse. The key label is *Syllabus*, which comprises the entire academic discourse around the specialized subject. The other three labels are *Interaction* (exchange of communication between the lecturer and students), *Digression* (a lecturer shifts to a more personal self and offers course-related asides) and *Other* (speech that cannot be classified under any of the other three labels and usually refers to the overall functioning of delivering the class and to non-course-related matters) [7].

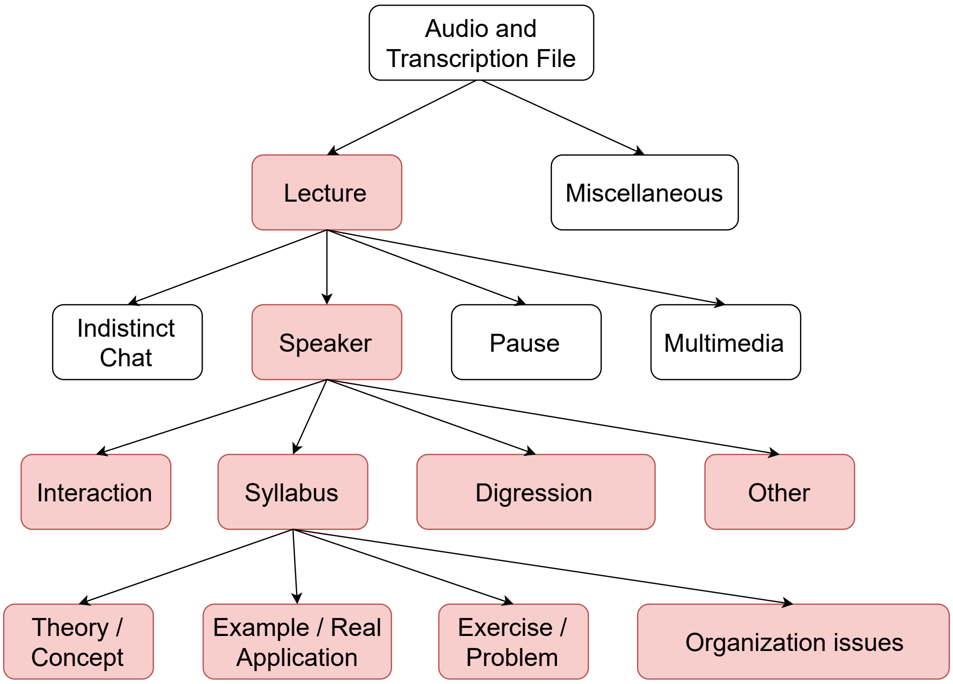


Figure 3.2: Label hierarchy of an audio & transcription file

Level 4: content-based lecture structure. This is the level that includes the phases of a regular expository class around the syllabus of a subject, namely *Theory/Concept*, *Example/Real Application*, *Exercise/Problem* and *Organizational issues*. Two observations that are worth mentioning:

- We include the label *Exercise/Problem* because our dataset comprises recordings of scientific/technical subjects. This label can however be ignored in the case of humanities and social science subjects.

- We include the label *Organizational issues* under *Syllabus* because providing general course information about schedules, teaching practice or policy grading is of interest for the carrying out of the syllabus. Nevertheless, we could also regard the label Organization as a sub-category of Speaker rather than Syllabus if we assume that students generally put much attention when the lecturer talks about organization matters.

All in all, after analyzing the class video recordings from VideoApuntes, we came up with **seven specific labels** (the seven colored leaf nodes in Figure 3.2) that enables us to split a lecture into segments and classify the contents of each segment. The selection of these labels is also supported by the common features used to categorize a university lecture in the literature (see, for example, [3].).

3.4.2. Segmentation and Labeling Procedure

This section explains the procedure we used to perform the data segmentation of the transcription file and the labeling of the segments under one of the seven classes discussed in the preceding section.

First and foremost, we segmented the transcription by identifying situations which pointed to a context switch and we determined whether said context change was associated to a change in the professor's discourse or not. Figure 3.3 is an example of a segment and Figure 3.4 shows the segment that comes right after the one in Figure 3.3.

> ***bueno entonces cómo van las correcciones esta tenéis hecha a hacer una corrección no lo hago a porque está bien claro ya o sea ya en las condiciones ejercicio pone si se va por decir vale es poco que decir como bueno a decir porque haya un decimal tomando como bueno vale está ahí en las condiciones [...] te pone probabilidad y tú has puesto ahí un porcentaje***

Figure 3.3: Example of a segment.

> ***bueno veréis que hay que hay una segunda tanda de poder subir los trabajos bueno la primera tanda aunque a la un segundo aunque alargamos el el momento de subir el ejercicio desde el viernes a las tres hasta el sábado a las tres solamente cinco alumnos subieron ese intervalo solamente cinco alumnos***

Figure 3.4: Segment that comes after the segment in Figure 3.3.

As we can see, both segments start with the word *bueno*, which generally implies a context switch. In the first segment, the lecturer talks about the correction of an exercise with the students. Figure 3.4 is the next segment after the one in Figure 3.3 and it also starts with the word *bueno*. In this case, the lecturer is talking about due dates of homework.

The labeling or classification of segments was performed by two members of the research group including the author of this work. We used a simple procedure to guide us when labeling: read the transcription segment and listen to the audio file until a change in context, tone or intonation was detected. This way we were able to determine when a section of audio was considered its own segment and we classified it by deciding the label that best fit the segment from the seven shown in the colored leaf nodes in Figure 3.2. Later, the discrepancies encountered between the two researchers were reviewed by the group supervisor and agreed by consensus.

Once data segmentation and labeling of the audio and transcription files were finished, we selected a group of people external to the project to label small fragments of audio files in order to ensure there was no bias in our labeling. The people we reached out to had no computer science related background and were as diverse as possible in regards to their professional background and age.

During labeling we used *Audacity* [21] as a guidance tool. *Audacity* is an open-source audio editor that enables to export a label track with our annotations, which, in our case, we used to label each section of the audio file. Audacity turned out to be a very helpful tool because it allowed us to export each label with time stamps automatically.

3.5 Label Description

The objective of this section is to describe each of the labels that appear in Figure 3.2. There are two main categories to take into account, the labels that do not have a transcription associated with them, which are out of the scope of the natural language analysis, and those that do have a transcription to work with. The first group of labels are the ones that belong to level 2 of Figure 3.2; that is, *Indistinct Chat*, *Pause* and *Multimedia*. The second group of labels are those of level 3 and level 4 in Figure 3.2.

3.5.1. Labels at level 2

An ***Indistinct Chat*** situation arises when there are background noises and the speaker (the lecturer) is not talking. We considered a segment of audio as *Indistinct Chat* when the speaker is indistinguishable but we can appreciate one or more people speaking and the content is inaudible and/or individual words are not identifiable.

A ***Pause*** is a situation that occurs when the lecturer pauses momentarily their discourse or talk. Segments of audio that are typically labeled as *Pause* include periods of silence between two other labels or situations in which the lecturer is performing an action other than speaking, for example, erasing the board, opening a slide presentation or handling the computer.

A ***Multimedia*** label appears when the audio & transcription file includes an external recording. *Multimedia* is regarded as a period of time during which the lecturer plays some type of external piece of audiovisual content such as, for example, a video. A *Multimedia* section of a class is not associated to a transcription since the lecturer is not the main speaker.

3.5.2. Labels at levels 3 and 4

The subtree rooted at *Speaker* depicts two hierarchy levels (levels 3 and 4 in Figure 3.2):

- Labels at level 3 denote the different types of lecturer speech that in turn determine a different kind of interpersonal relationship between the lecturer and the students,
- Labels at level 4 focus exclusively on the lecturer discourse and characterize the syllabus content,

In the following, we detail the labels at level 3, *Interaction*, *Digression* and *Other*, and subsequently we present the breakdown of the label *Syllabus*.

Interaction

This is a very common situation that arises when the professor talks, asks, or interacts with the students. A situation of lecturer-student interaction arises when a lecturer forwards a question to the audience or when a student raises a question about the contents of the lecture.

We regard *Interaction* as a part of the transcription that may be interspersed by pauses (a pause in the speaker's speech may involve a student asking a question) and that forwards a question or an answer directly to someone. In addition, responses to questions may come preceded by words like *yes, no, sure, not quite*, etc. Table 3.3 shows the transcription of an *Interaction* label and a more human readable version (correction).

Transcription	lo me dejas acabar me dejas me dejas me dejas el pero me dejas que lo explique vale gracias evelyn
Correction	¿Me dejas? ¿Me dejas que lo explique? Vale, gracias Evelyn.

Table 3.3: Sample of an *Interaction* transcription and its correction.

Digression

Digressions allow the lecturer to offer students course-related asides and they are commonly used to lighten up the content of the lecture, for example, self-mention or joking [7]. Digressions take the form of personal comments or anecdotes and help create a relaxed environment and maintain a positive lecturer–audience relationship.

A digression arises when the lecturer makes a comment or gives an opinion about something related to the lecture exposition. We will use the *Digression* label for comments made by the lecturer related to the lecture exposition and based on an experience or a particular circumstance. An example is shown in Table 3.4, sometimes transcriptions are directly human readable.

Transcription	Sí al principio os parece un poco raro pero es lo que hay nos tenemos que ir acostumbrando.
Correction	Sí, al principio os parece un poco raro pero es lo que hay, nos tenemos que ir acostumbrando.

Table 3.4: Sample of a *Digression* transcription and its correction.

Other

The *Other* label is the category where we classify anything the lecturer says that is out of the scope of all of the other labels. It is, in some sense, the juxtaposition of the course-related asides label; that is, it is a situation that occurs during the lecture but it is unrelated to it. As an example of *Other*, we can mention a situation in which the lecturer asks the students if the temperature of the classroom is adequate so as to regulate the heating. Table 3.5 shows the transcription of an *Other* label and a more human readable version.

Transcription	Por, la hoja de firmas por favor que pase, vale.
Correction	Por favor, la hoja de firmas, por favor, que pase, vale.

Table 3.5: Sample of a *Other* transcription and its correction.

The *Syllabus* label encompasses labels that represent the academic discourse of the lecturer to teach the contents of the matter. All the labels under *Syllabus* identify a single speaker, the lecturer, and their speech is usually fluid and concise.

Syllabus::Theory/Concept

The *Theory* labels are the most common, they arises when the lecturer is explaining something from the syllabus. We identified as *Theory* those sections of audio where the professor would introduce a concept and then explain what it is or what it means.

Transcription	el timer dos tiene un pre divisor y un post divisor este no se utiliza en la modulación de ancho de pulso la salida se obtiene de aquí de la serie al timer dos por tanto en la ecuación no aparece
Correction	El timer 2 tiene un predivisor y un postdivisor. Este no se utiliza en la modulación de ancho de pulso. La salida se obtiene de aquí, de la serie al timer 2, por tanto en la ecuación no aparece.

Table 3.6: Sample of a *Theory/Concept* transcription and its correction.

Table 3.6, the lecturer introduces the concept of the difference between two matters and immediately explains what the difference is.

As we can see in Table 3.6, many times the difference between the raw transcription and the human readable version is just a matter of punctuation.

Syllabus::Example/Real Application

An example is a lapse of time during a lecture where the professor is illustrating or exemplifying the concepts being taught. We shall consider an *Example/Real Application* when the lecturer uses a real-life application or a circumstance to explain the theory or concept previously explained. Table 3.7 shows the transcription of an *Example/Real Application* label and a more human readable version.

Transcription	por ejemplo la probabilidad de que salga el seis en un dado no trucado de acuerdo a el ocho en un dado no trucado
Correction	Por ejemplo, la probabilidad de que salga el 6 en un dado no trucado, de acuerdo, el 8 en otro dado no trucado.

Table 3.7: Sample of a *Example/Real Application* transcription and its correction.

Syllabus::Exercise/Problem

A *Problem* label can be regarded as those where the professor is applying the knowledge previously explained into a practical situation that could be taken into account on a real-life application. Since we are labeling science-based subjects, it is easier to detect these kind of segments because they normally include numbers and formulae. Table 3.8 shows the transcription of an *Exercise* label and a more human readable version.

Transcription	En este caso la resistencia a ese R, la que vamos a diseñar a calcular el valor para el cual no se superan la potencia máxima de en el diodo. Vale
Correction	En este caso la resistencia es R, la que vamos a diseñar, calcular el valor para el cual no se supera la potencia máxima del diodo.

Table 3.8: Sample of a *Exercise/Problem* transcription and its correction.

Syllabus::Organization Issues

The *Organization Issue* label happens when the lecturer talks about other course-related activities such as lab sessions, assignments or exams. It refers as a part where the main focus of the matter has some relation with the subject but not with the lecture *per se*. Table 3.9 shows the transcription of an *Exercise* label and a more human readable version.

Transcription	vale a ver poliformat este ratón greg resolución es mi poliformat ing esto ha cambiado a ver ana porque me ha cambiado bueno nada asignaturas matemáticas todos vale hoy ha cambiado esto muy bien
Correction	Vale, a ver PoliformaT. Esto ha cambiado. A ver, Ana. ¿Por qué me ha cambiado? Asignaturas, Matemáticas, todos, ¿vale? Hoy ha cambiado esto. Muy bien.

Table 3.9: Sample of a *Organizational Issues* transcription and its correction.

In addition to the data shown in Table 3.2, Table 3.10 depicts a division between genders in the percentage of time spent on each category or label of the hierarchy tree in Figure 3.2 in lecture.

Label Name	Male %	Female %
Theory/Concept	32.8	12.1
Exercise/Problem	12.2	21.0
Example/Real Application	16.6	7.0
Organization Issues	7.9	5.9
Interaction	9.8	23.3
Digression	3.2	0.9
Other	0.7	1.7
Indistinct Chat	0.9	2.8
Pause	2.4	7.0
Multimedia	0.0	0.0
Miscellaneous	13.6	18.4

Table 3.10: Percentage of occurrences of each label in relation to the total number of minutes per gender.

Finally, we would like to stress that our aim focuses on the first seven labels, each denoting a different **teaching activity** deployed by the teacher to create the conditions for learning, stimulating conceptual thinking, promoting discussion and engaging students in critical thinking and problem solving. Ultimately, all lecturers aim to creating the proper atmosphere that facilitates the information transmission in an interactive environment.

CHAPTER 4

Text classification

In this chapter we will show the design, development and performance of our classification model. Since our model builds on top of XLM-RoBERTa, the first section of this chapter is devoted to explaining the pre-trained multilingual language model XLM-RoBERTa; i.e., the default characteristics that XLM-RoBERTa brings in and that we will exploit to desing our classification model. Section 4.2 details the design choices of our proposed model. At the end of this chapter, we report the results obtained with our model.

4.1 Pre-trained XLM-RoBERTa model

Among recent state-of-the-art NLP models, there is a significant trend for Transformer-based models as discussed in section 2.2. The vast majority of the work on Transformer-based models put the focus on monolingual models, mostly on English language, whereas in our case we need a model trained in Spanish.

Out of the pool of eligible models, XLM-RoBERTa stands out as a cross-lingual model trained in 100 languages that achieves a performance comparable to monolingual models in a variety of tasks such as named entity recognition, question answering, sentiment analysis, natural language inference, etc.

The XLM-RoBERTa model has 12 Transformer layers, each composed of 768 hidden-state units, 3072 feed-forward hidden-state units and 8 self-attention heads. The model has approximately 270M parameters and a vocabulary size of 250k tokens.

XLM-RoBERTa uses SentencePiece [22], an unsupervised, fast and easy to train text tokenizer and detokenizer which does not depend on language-specific pre/postprocessing. Besides it deals with the *whitespace problem* (tabs, spaces, new lines, etc.) by treating it as a symbol since some languages like Chinese or Japanese do not use white spaces to divide words. The SentencePiece tokenizer serves to translate the input text into a list of numerical identifiers that can be further processed by the model. Specifically, the tokenizer learns the *subwords* and assigns them a unique identifier, wherein the most common words are categorized as a whole subword and the least common words are split into multiple subwords. Thus, words such as articles, root words, prefixes and suffixes are usually categorized as subwords. For example, the word *unknowingly* would be commonly partitioned into subwords like this: *[un] [know] [ing] [ly]*.

The tokenizer also adds some special tokens such as the *beginning of sequence* token and *end of sequence* token at the beginning and ending of the sequence, respectively. If padding is used, the tokenizer appends a *pad* token after the *end of sequence* token until the length of the segment equals the specified length. The model also receives an attention mask that indicates which tokens are words and which ones are pad tokens. The attention mask is a list of 0's and 1's that is used when the model is performing self-attention to ignore the pad tokens.

The XLM-RoBERTa model is pre-trained on 2.5 TB of clean CommonCrawl data in 100 languages using a combination of Masked Language Modeling and Next Sentence Prediction:

- *Masked Language Modeling* (MLM) is a task that consists in filling out blanks in a sequence. Specifically, given a random token in an input sequence (mask token), a model predicts (reconstructs) the masked word using the context words surrounding the mask token.
- *Next Sentence Prediction* (NSP) is a technique used during pre-training to model relationships between sentences. Given a sequence of pairs of text segments as an input, NSP is used to train a classifier telling whether the second segment of the pair is a direct successor of the first one or not. Instead of using the *beginning of sequence* token, NSP uses a *classifier token*, which is used when doing sequence classification (classification of the whole sequence instead of per-token classification). Thus, NSP trains the model to learn a sequence representation in the embedding of the *classifier* token.

4.2 Classification model

In this section we will explain the NN-based model built on top of XLM-RoBERTa to learn classifying academic transcription segments into the seven classes previously discussed in chapter 3. In order to use XLM-RoBERTa as a classifier, we need to make some changes in the tokenization process as well as in the XLM-RoBERTa architecture before training the model with our dataset.

4.2.1. Tokenization

We apply the tokenization process to a given input transcription segment (sequence) adding the *classifier* token used in NSP instead of the *beginning of sequence* token. As explained in the above section, when doing sequence classification, the embedding of the *classifier token* is a representation of the whole input sequence, thus we use it to classify the input transcription segment.

We set the maximum sequence length to 512 tokens, which is the maximum length supported by XLM-RoBERTa, because the longer the sequence the easier to classify a segment thanks to the larger amounf of context information comprised in it. If the segment contains fewer than 512 tokens we apply padding. For sequences longer than 512 tokes we split them using a *sliding window* with a *stride* of 0.8 * *max_seq_length* (410 tokens). All this is done with the purpose of having all segments of the same size.

Following, we present an illustrative example that shows the tokenization process depicted in Figure 4.1.

The raw transcription text shown in the upper excerpt of Figure 4.1 is our working segment. First, the tokenizer divides the whole segment into subwords (the result is shown in the middle excerpt of Figure 4.1). XLM-RoBERTa's tokenizer has a vocabulary of 250.000 subwords learned from 100 languages. Among these subwords, a small part is reserved to special tokens such as the *beginning of sequence* token, the *padding* token, etc. As we can see in the middle excerpt of Figure 4.1, some words like *'el'*, *'dos'* and *'este'* have their own subword while *'divisor'*, *'modulación'* and *'ecuación'* are split into multiple

Raw text:
el timer dos tiene un pre divisor y un post divisor este no se utiliza en la modulación de ancho de pulso la salida se obtiene de aquí de la serie al timer dos por tanto en la ecuación no aparece el

Subword division:
′[CLS]′, ′_el′, ′_timer′, ′_dos′, ′_tiene′, ′_un′, ′_pre′, ′_di′, ′visor′, ′_y′, ′_un′, ′_post′, ′_di′, ′visor′, ′_este′, ′_no′, ′_se′, ′_utiliza′, ′_en′, ′_la′, ′_modul′, ′ación′, ′_de′, ′_an′, ′cho′, ′_de′, ′_puls′, ′o′, ′_la′, ′_salida′, ′_se′, ′_ob′, ′tiene′, ′_de′, ′_aquí′, ′_de′, ′_la′, ′_serie′, ′_al′, ′_timer′, ′_dos′, ′_por′, ′_tanto′, ′_en′, ′_la′, ′_e′, ′cu′, ′ación′, ′_no′, ′_aparece′, ′_el′, ′[EOS]′

Token ids:
0, 88, 21991, 655, 5904, 51, 479, 45, 51858, 113, 51, 1305, 45, 51858, 473, 110, 40, 24514, 22, 21, 17055, 4117, 8, 142, 3089, 8, 55111, 31, 21, 114823, 40, 995, 81228, 8, 9877, 8, 21, 6432, 144, 21991, 655, 196, 4104, 22, 21, 28, 1010, 4117, 110, 39426, 88, 2

Figure 4.1: Example of a tokenization process: (upper excerpt) raw transcription text; (middle excerpt) subword division; (bottom excerpt) identifiers of tokens.

subwords. This happens because, as we explained earlier, SentencePiece learns to assign a single subword to a common word, and multiple subwords to uncommon words.

We can also observe there are two special tokens, represented as *′[CLS]′* and *′[EOS]′*, which correspond to the *classifier* token and the *ending of sequence* token, respectively. The bottom excerpt of Figure 4.1 shows that each subword has a unique token id, 0 being the id of the *classifier* token and 2 the id of the *ending of sequence* token. Since the length of the segment is less than 512 tokens, we apply the *padding* token, whose id is 1, until completing the specified length. The resulting list of integer-valued ids is the actual input that XLM-RoBERTa receives.

4.2.2. Tokenized dataset

As we explained in sections 3.4 and 3.5, our dataset is composed of manually labeled transcriptions obtained from academic lectures. We identified seven relevant classes: *Theory/Concept*, *Exercise/Problem*, *Example/Real Application*, *Organization Issues*, *Interaction*, *Digression* and *Other*.

Table 4.1 shows the composition of our dataset by class or label. For each label, we report:

- the number of segments,
- the total number of tokens ,
- the average token length of the segments ,
- the maximum length in tokens of a segment.

As we can see in Table 4.1, our dataset is rather imbalanced because some classes like Theory/Concept or Exercise/Problem are much more frequent than other classes like Digression or Other. This is reasonable, as it is in line with the nature of the academic discourse, wherein the largest part of the teacher′s speech is devoted to developing the

Label	Num Segments	Total Tokens	Avg. Tokens	Max Tokens
Theory/Concept	454	115885	255.25	3019
Exercise/Problem	537	77866	145.01	1481
Example/Real Appl.	347	63239	182.24	1845
Organization	260	37083	142.63	1989
Interaction	567	66326	116.98	3878
Digression	118	11857	100.48	647
Other	112	5416	48.36	297
Total	2395	377672	157.69	3878

Table 4.1: Distribution of our data by label. The number of tokens was obtained by using XLM-Roberta's tokenizer.

contents of the syllabus of the subject. As a result, the total number of tokens of the most populated classes is obviously higher.

Looking at the figures in Table 4.1 some conclusions about the composition and structure of each class can be drawn. For example, we can observe that , token-wise, the *Other* class is less than 5% in size than the *Theory/Concept* class . However, the *Other* class is fairly regular as it is mostly composed of relatively short segments. This is indicated by the lowest value of the maximum number of tokens (297) in a segment and also by the lowest average number of tokens (48.36). In contrast, the *Interaction* class is highly irregular since it contains segments of variable length. If we look at the numbers in Table 4.1, we can see that *Interaction* is the class with the largest number of segments (567) and it is also the class with the longest segment (3878), while this class has about half the number of tokens of the *Theory/Concept* class (57.23%), and its average number of tokens is closer to the less populated classes.

4.2.3. Classification task

We used the pre-trained XLM-RoBERTa model explained in section 4.1 and we added a *classification head* on top of the last Transformer layer. In this section, we describe the architecture of the classification head, its input and its output. We also provide details on the training process and the libraries employed.

The classification head takes as input the sequence representation (segment) contained in the embedding of the *classifier token* ([CLS] in Figure 4.1). It consists of a dense layer of hidden size (768 units) with *tanh* activation function followed by a dense layer of seven units (one unit for each label/class associated to one academic activity) with *softmax* activation. The output of the classification head is a list containing the probability that the input segment belongs to each of the seven classes. We used the *Adam* algorithm with weight decay fix as optimizer and *categorical cross-entropy* as our loss function for fine-tuning.

Among the libraries and frameworks that were used to carry out our training process, we highlight two of them:

- HuggingFace's Transformers library [23] is a popular Python library that offers state-of-the-art pre-trained NLP models for Pytorch and Tensorflow 2.0. Particularly, we obtained the pre-trained XLM-RoBERTa model, known as *xlm-roberta-base*, from the HuggingFace's Transformers repository.

- SimpleTransformers [24] is a NLP library built on the HuggingFace's Transformers library that helps simplify the usage of Transformer models without compromising on utility. SimpleTransformers simplifies the usage of HuggingFace's Transformers in a similar way as Keras does with Tensorflow.

4.3 Experimental evaluation

In this section we present the implementation details of our classification model, the evaluation metrics used to assess its performance and the results obtained as well as a thorough analysis of the results.

4.3.1. Setup configuration

We trained our model with two computers: a Nvidia Titan V and a Nvidia Geforce RTX 3090. Although we used powerful graphics cards, we experienced problems with memory usage as the model needs many GB of GPU memory. Consequently, we were forced to restrict our experiments to smaller batch sizes, but we were able to simulate bigger batch sizes using the gradient accumulation steps hyperparameter, which controls the number of required steps to update the model weights. Hence, the simulated batch size is equal to the batch size multiplied by the gradient accumulation steps hyperparameter. The results we report below were obtained with the Nvidia Geforce RTX 3090.

We used the *Weights and Biases* framework [25] to tune our hyperparameters according to the model performance using Bayesian Optimization with the two Nvidia Titan V. Our final hyperparameters are set as follows:

- a learning rate equal to 0.00005,
- 40 epochs,
- batch size of 8 segments (due to memory constraints),
- gradient accumulation steps of 32 (for a simulated batch size of 256 segments),
- weight decay of 0.0007.

Table 4.2 depicts the composition of the dataset we used for the experimentation. Each row shows the details of one out of the 27 transcription files. The first three columns indicate, respectively, the lecturer, the course name and whether the transcription was manually corrected. The rest of the columns are:

- the number of segments comprised in the transcription file,
- the index number of the first and last segment of the transcript file,
- the dataset is partitioned in 10 equal subsamples for training and testing (see details in the following section); the last column shows the number of the partition or the numbers of the two consecutive partitions in which the segments of the file fall in.

Prof.	Course	Manually reviewed	Number of segments	Index of segments	Partition
1	Statistics	No	104	0 - 103	1
1	Statistics	No	141	104 - 244	1(240) - 2
1	Statistics	No	101	245 - 345	2
2	Mathematics	No	73	346 - 418	2
2	Mathematics	No	141	419 - 559	2(479) - 3
2	Mathematics	No	105	560 - 664	3
2	Mathematics	No	92	665 - 756	3(718) - 4
3	Oceanographic Physics	No	60	757 - 816	4
3	Oceanographic Physics	No	84	817 - 900	4
3	Oceanographic Physics	No	60	901 - 960	4(958) - 5
4	Networks & Teledetection	No	62	961 - 1022	5
4	Networks & Teledetection	No	54	1023 - 1076	5
4	Networks & Teledetection	No	40	1077 - 1116	5
4	Networks & Teledetection	No	71	1117 - 1187	5
5	Microprocessed Systems	No	43	1188 - 1230	5(1198) - 6
5	Microprocessed Systems	No	64	1231 - 1294	6
5	Microprocessed Systems	No	38	1295 - 1332	6
6	Electronic Devices	No	110	1333 - 1442	6(1437) - 7
6	Electronic Devices	Yes	110	1443 - 1552	7
6	Electronic Devices	Yes	159	1553 - 1711	7(1676) - 8
7	Mathematics	Yes	65	1712 - 1776	8
7	Mathematics	No	46	1777 - 1822	8
7	Mathematics	Yes	119	1823 - 1941	8(1916) - 9
7	Mathematics	No	70	1942 - 2011	9
8	Digital Signal Treatment	Yes	193	2012 - 2204	9(2156) - 10
8	Digital Signal Treatment	Yes	41	2205 - 2245	10
8	Digital Signal Treatment	No	149	2246 - 2394	10

Table 4.2: Dataset composition.

4.3.2. Evaluation metrics

We evaluated our classification model with the dataset whose composition is exposed in Table 4.2 and the setup configuration described in the preceding subsection. The purpose of the evaluation is to measure the performance of the model when classifying segments into one of the seven classes: (1) *Theory/Concept*, (2) *Exercise/Problem*, (3) *Example/Real Application*, (4) *Organization issues*, (5) *Interaction*, (6) *Digression* and (7) *Other*.

The evaluation metrics we used to assess the performance of the classifier are *accuracy*, *precision*, *recall* and *F-score*. Firstly, we will explain the concepts of true positives, false positives, true negatives and false negatives, which appear in the formulae that model the metrics:

- True Positives (TP): samples (segments) that are labeled as class X and that the model correctly classifies into class X.

- False Positives (FP): samples (segments) that are not labeled as class X but the model classifies into class X.

- True Negatives (TN): samples (segments) that do not belong to class X and the model correctly classifies as not belonging to class X.

- False Negatives (FN): samples (segments) that belong to class X but that the model incorrectly classifies as not belonging to class X.

Now, having N being the total number of samples (2395 in our dataset), and C the set of classes (C = {*Theory/Concept*, *Exercise/Problem*, *Example/Real Application*, *Organization issues*, *Interaction*, *Digression* and *Other*} and |C| = 7), the accuracy is defined as the average number of correct predictions over all classes:

$$accuracy = \frac{1}{N}\sum_{c \in C} TP_c$$

Since the accuracy is a global measure of the performance of the model, it gives us no information about the performance in each class. Furthermore, accuracy may not be a good metric in imbalanced datasets, as it may be highly influenced by the correct classification of the most populated classes in the dataset. In problems where there is a large class imbalance, a model that predicts the value of the majority class achieves a high classification accuracy but may not be useful in the problem domain. For example, let us suppose that we build a dataset with all the samples of *Digression* (118 samples) and *Interaction* (567 samples). Now, say that we train a classifier and it turns out to be heavily biased and classify all samples as *Interaction*. This classifier would have an accuracy of 0.827 while it did not learn to identify *Digression* at all. Because of this reason, we will also use other metrics such as precision, recall and F-score.

Precision of a class c is defined as the fraction of samples that belong to c among all the samples the model classifies as belonging to such class (also known as positive predictive value). Recall (also known as sensitivity) gives us the measure of how our model correctly identifies True Positives, i.e., the fraction of samples correctly classified as belonging to class c among all the samples of such class.

$$precision_c = \frac{TP_c}{TP_c + FP_c} \qquad recall_c = \frac{TP_c}{TP_c + FN_c}$$

Finally, F-score of a class is the weighted harmonic mean of the precision and recall of the class. We report a balanced F-score, meaning that we equally weighted precision and recall with the following formula:

$$F\text{-}score_c = \frac{2 \cdot precision_c \cdot recall_c}{precision_c + recall_c}$$

By using the F-score, we can evaluate the performance of our model for each class, as a high F-score requires both high precision and recall, otherwise the resulting F-score would be greatly affected. For example, a low precision and a high recall values for a class means the model is unable to correctly identify the samples that belong to the class. Conversely, a high precision and low recall for a class means that our model is biased toward that class.

4.3.3. Results

Table 4.3 shows the results of **accuracy** obtained with a 10-fold cross-validation (data are split in 10 folds or partitions). The meaning of each column is as follows:

- first column shows the results without shuffling the data; the original partitions that result before shuffling the data as shown in Table 4.2,
- the second column shows the results obtained when segments are shuffled randomly before splitting them in 10 partitions,
- the last column is the stratify 10-fold cross-validation; i.e, we shuffled the dataset and divided it into ten partitions ensuring that each fold has the same proportion of segments of each class.

On each iteration, one partition is taken as as a hold out or test data set and the remaining nine partitions as training data set.

	Accuracy		
Iteration	without shuffling	with shuffling	with shuffling + stratify
1	0.72803	0.74895	0.81250
2	0.72385	0.76151	0.73750
3	0.66109	0.74059	0.74583
4	0.72385	0.76569	0.72083
5	0.69874	0.75314	0.78333
6	0.73222	0.73640	0.79916
7	0.71129	0.74059	0.71548
8	0.68619	0.72803	0.73221
9	0.90795	0.76987	0.74895
10	0.93443	0.67213	0.78242
Average	0.75076	0.74169	0.75782
Variance	0.00772	0.00070	0.00104
Standard Deviation	0.08787	0.02646	0.03225

Table 4.3: Results obtained with 10-fold Cross-Validation with and without shuffling the data, and using stratify.

An interesting result in Table 4.3 is that the accuracy values of iterations 9 and 10 when using non-shuffled data is $\approx 90\%$, which are significantly higher than the rest of accuracy values. The reason for these high values is given by two factors: (1) the manually reviewed transcriptions and (2) the sensitivity of the model to unseen or rarely seen courses/professors during the training phase. On the one hand, the quality of the manually reviewed transcriptions is obviously closer to written text, which is the kind of text XLM-RoBERTa was trained with. On the other hand, the way the segments are distributed in partitions 9 and 10 implies that there are enough manually reviewed segments of those courses/professors outside partitions 9 and 10. These factors lead to the observed increase in accuracy.

To illustrate how the distribution of the segments impacts the performance of the model, let's focus on, for example, the ninth iteration. During this iteration, the model is trained with partitions 1 to 8 and partition 10, and the model is evaluated with partition 9. The evaluation of the model with partition 9 involves evaluating with (see Table 4.2):

- a small section of a manually reviewed Mathematics transcription from segment 1917 to segment 1941,
- a full class of an automated transcription of Mathematics from segment 1942 to segment 2011,
- a small section of a manually reviewed class of the course Digital Signal Treatment from segment 2012 to segment 2156.

During training, the model sees two lectures from the Digital Signal Treatment course, one of them being manually reviewed, and almost three Mathematics lectures from the same professor, two of which are manually reviewed. Hence, the training data happen to be very favourable when evaluating partitions 9 and 10, because they include transcriptions of the same course, same lecturer and same quality as those being evaluated later.

The worst results without shuffling are obtained when evaluating the model with partitions 3, 5 and 8. In these evaluations, the model does not see a sufficiently significant portion of data during training. For example, when evaluating partition 5, the model does not see any segment belonging to professor 4 or to the Network & Teledetection course during the training phase.

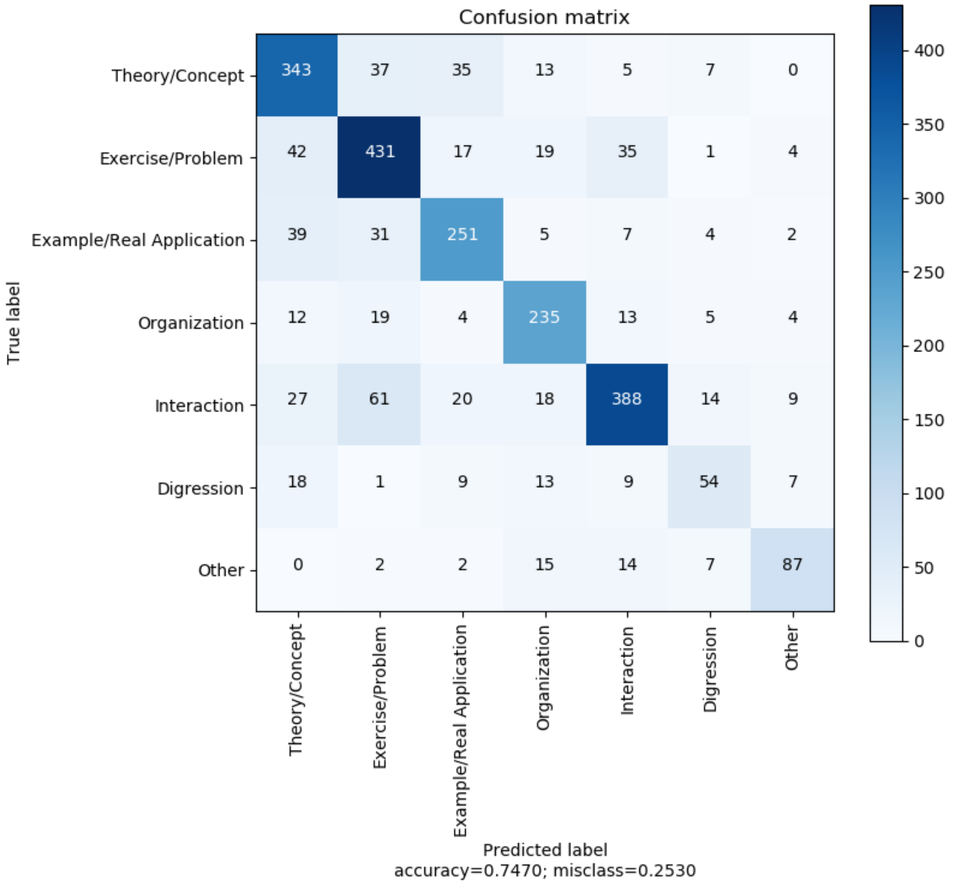
Comparing the results obtained with and without shuffling, we can see that the average accuracy drops by around ≈1%. Even though the results without shuffling may suggest that the model is sensitive to the training data, the results in the second column of Table 4.3 seem to indicate otherwise. That is, the model turns out not to be as data-sensitive as the figures in the first column might in principle indicate.

The results obtained with the stratified 10-fold cross-validation are shown in the third column of Table 4.3. We obtained an average accuracy of 75.78%, with a standard deviation of 3.23% and a variance of 0.104%. We see that the accuracy across iteration varies between 71% and 81%. Overall, the average accuracy of the stratified cross-validation is slightly better than those obtained with the without-shuffling strategy, while the variance and deviation are closer to those of the with-shuffling strategy. This confirms that a uniform distribution of classes in each fold yields a slightly superior performance over the other two approaches.

Label	**Precision**	**Recall**	**F-Score**
Theory/Concept	0.713	0.780	0.745
Exercise/Problem	0.741	0.785	0.762
Example/Real Appl.	0.742	0.740	0.742
Organization	0.739	0.805	0.770
Interaction	0.824	0.723	0.770
Digression	0.587	0.486	0.532
Other	0.769	0.685	0.725

Table 4.4: Precision, Recall and F-Score by class.

We also report the values of precision, recall and F-score for each class in Table 4.4. We can observe that the metrics of the *Digression* class fall behind the rest of classes. This can be due to several factors, but we believe this is mainly caused by the low number of



	Theory/Concept	Exercise/Problem	Example/Real Application	Organization	Interaction	Digression	Other
Theory/Concept	343	37	35	13	5	7	0
Exercise/Problem	42	431	17	19	35	1	4
Example/Real Application	39	31	251	5	7	4	2
Organization	12	19	4	235	13	5	4
Interaction	27	61	20	18	388	14	9
Digression	18	1	9	13	9	54	7
Other	0	2	2	15	14	7	87

Figure 4.2: Aggregated Confusion Matrix of 10-fold Cross-Validation with shuffling and stratify[1].

samples of this class in our dataset and the difficulty we experienced to correctly label this class.

It is also noticeable that the precision value of the *Interaction* class is somewhat higher than its recall and also higher than the precision of the other classes. This occurs because the number of False Positives is relatively low compared to that of False Negatives for this class; that is, the model has some difficulty in identifying the *Interaction* class.

Figure 4.2 shows the confusion matrix where rows show the true label of the segments, i.e., the label we manually assigned to the segments, and columns represent the prediction of our model. The values on the diagonal of the confusion matrix indicate how many segments are correctly classified for each class, while the rest of the confusion matrix shows the misclassified segments.

Let us now turn our attention to the False Positives, i.e., the values in the columns of the confusion matrix. We can see that the majority of high values in the columns other than the diagonals are concentrated in three classes: *Theory/Concept*, *Example/Real Application* and *Exercise/Problem*. We can thus say that our model has a certain bias towards this group of classes which otherwise is reasonable since these three classes are the main representatives of the academic discourse of the lecturer and share a substantial part of

[1] The accuracy reported in the confusion matrix differs from the result reported in Table 4.3 because the confusion matrix was generated in a different execution under the same conditions, but there is a certain level of unavoidable randomness that makes it difficult to get the exact same results again.

their vocabulary. Additionally, more than half of the dataset segments belong to one of these three classes.

It is also worth mentioning that there is a significant amount of misclassifications between the *Exercise/Problem* and *Interaction* classes. This is due to the fact that Interactions between student and lecturer typically come up during the solving of problems and exercises: either a student asking for clarification or the lecturer asking the students.

The values of the *Digression* row indicates a low recall for this class since the values are relatively high in relation to the number of samples correctly classified (54). This means that a large number of False Negatives are found for this class and show the difficulty of the model to correctly classify segments that are labeled as *Digression*. This observation about the *Digression* class in the confusion matrix is consistent with the value of recall shown in Table 4.4.

The *Other* class column reveals a high precision value whereas, as in the case of the *Digression* class, the values in the *Other* row indicate a lower recall value. Interestingly, the *Organization* class shows the opposite behaviour: a high recall and a lower precision value (low values in the row in comparison to the values in the column). This means the model is pretty successful in correctly classifying many of the segments labeled as *Organization* but it also tends to classify as *Organization* segments that do not actually belong to this class. The high recall may be explained because the *Organization* class comprises a particular vocabulary that distinguishes it from other classes. We will analyze this aspect in the next chapter.

4.3.4. Conclusions and further improvements

We can conclude that our model obtains satisfactory results despite the low number of labeled samples for this type of NLP task. We have also observed that:

- the group of academic classes (*Theory/Concept*, *Example/Real Application*, *Exercise/Problem*) concentrates a large part of the errors because this group of classes makes up more than half of the dataset and the classes within it share a large part of their vocabulary among them,
- there is some confusion between the *Interaction* and *Exercise/Problem* classes which happens because many interactions student-lecturer take place during examples or exercises in class,
- the model achieves good results for the *Organization* class, probably because this class has a distinctive vocabulary (dates, grading, etc.) and it was also the easiest class to identify during labeling.

Lastly, we propose some improvements to increase the performance of our model:

- Increasing the dataset size: as shown in [15] and [17], increasing the amount of data used for training leads to a noticeable increase in performance. We expect to come up with an automated segmentation process and use our classification model to help us augment the dataset.
- Employing a Language Model to correct and spellcheck the automatic transcriptions. Although the quality of the automatic transcriptions is high, we believe that the model could perform better if we increase the quality of the transcriptions, so that it approaches the quality of written text,

- Employing XLM-RoBERTa-large instead of XLM-RoBERTa-base. With the large version of XLM-RoBERTa, we should obtain a better performance since it outperforms the base model at the cost of a more expensive training in terms of memory consumption and training time.

CHAPTER 5
Human-like Classification

The objective of this additional analysis is to compare the results obtained with the Neural Network model versus a more human-like classification. The question we want to answer is "would a person classify a particular segment into the same class as the neural network model?". The design of the human decision relies on a classification tree that compares the vocabulary shared by a segment and a teaching activity (class or label).

5.1 Introduction

One first observation is that classes *Theory*, *Exercise* and *Example* share a great deal of vocabulary as these are the three key teaching activities to attain the informational transmission of the subject. We also saw in the preceding chapter that most of the classification errors of the NN model are concentrated on these three classes. Taking into account that even for a knowledgeable person it would be hard to distinguish among the three classes, we decided to group them under a single teaching activity. Consequently, the purpose of the human classification is recognizing a teaching activity among *Theory-Exercise-Example*, *Organization*, *Interaction*, *Digression* and *Other*.

Our objectives in this chapter are:

- creating a vocabulary for each of the five aforementioned teaching activities,
- creating a vocabulary for the segment to classify,
- design a decision tree that that follows a human-like decision process,
- comparing the obtained results with those obtained with the neural network model.

5.2 Creation of a vocabulary

We created a key vocabulary for the five activities (*Theory-Exercise-Example*, *Organization*, *Interaction*, *Digression* and *Other*) as well as a custom vocabulary for *Organization* and *Interaction*. The same process used to create the vocabulary of the teaching activities was applied to create the vocabulary of a segment.

To create a vocabulary out of a set of segments, we iterate over all the transcriptions and create a list with the words comprised in them. We used a bag-of-words representation to create a vocabulary and to avoid repetition of words.

A **bag-of-words (BoW) representation** is a simplifying strategy used in NLP. In this model, a text is represented as the bag (multiset) of its words, without taking into account grammatical structure or the order of terms but retaining multiplicity. We will use the BoW representation in all the sections described in this chapter. Figure 5.1 is an example of a bag of words taken from the transcriptions. It shows a word and its number of occurrences. In this example, the word ***que*** appeared 13595 times, the word ***de*** 11565 times, etc.

{'que': 13595, 'de': 11565, 'la': 9932, 'el': 7574, 'y': 7469, 'a': 7221, 'es': 6872, 'en': 5452, 'no': 5154, 'lo': 4846, 'por': 3709, [...],'incorrectas': 1, 'dijeras': 1 }

Figure 5.1: Sample of a bag-of-words.

The problem with the BoW representation is that **stopwords** appear as the most frequent. Stopwords are words which do not contain enough importance to determine the key terms of the vocabulary. Since we do not want these words to take up space in our bag-of-words, we can remove them from our list. To this end, we opted to use the list of stopwords of the *Natural Language Toolkit* (NLTK) Python library [26] (NLTK further on).

Once stopwords are removed from the vocabulary, another problem arises: many words have the same root or stem but are counted as completely different words. To have a more accurate picture of the semantics of the text, we need to avoid this.

One approach to group words according to their root is to use a **stemming algorithm**. Stemming consists in removing prefixes and suffixes so that words with the same origin are counted as having the same root: e.g., words that end in *-ing*, *-ly* or *-ed*. This seems to be a better approach, but it still has some drawbacks. Two main disadvantages of this method can be identified: (1) stemming does not produce actual words as an outcome, which can cause uncertainty, for instance, if we stem the words ***university*** and ***universe*** they will, most likely, be stemmed into ***univers*** although they have a different meaning; and (2) many verbs have irregular conjugations, for example, the verb ***to be*** can be conjugated into ***I am***, ***you are***, ***she is*** etc which will not be stemmed into the same root.

A better approach to avoid this problem is to use **lemmatization**. Lemmatization follows a similar process to stemming but it involves resolving words into their lemmas, which are actual root words. In the previous example, using lemmatization with ***she is*** would result in the lemmas ***she*** and ***be*** which is what we are looking for.

5.2.1. Identification of relevant terms

As a first approach, our idea was to retrieve the **most frequent vocabulary** for each activity or label. That is, to create a *bag-of-words* of the vocabulary by applying the techniques mentioned in the preceding section and ordering the words from higher to lower frequency of appearance. However the *Most Frequent Vocabulary* has an important drawback, there are many frequent words that are not actually useful to identify which label the bag refers to. Some example of these words are ***si***, ***vale***, ***aquí*** or ***entonces*** etc. In order to identify the most characteristic terms of each activity we relied upon the **Term Frequency - Inverse Document Frequency** (TF-IDF). This metric assesses the relevance of a term in a given document. The formula for TF-IDF is:

$$w_{ij} = tf_{ij} \times \log\left(\frac{N}{df_i}\right)$$

where tf_{ij} represents the number of occurrences of the term i in document j, df_i is the number of documents containing the term i and N is the total number of documents.

The formula tells us that a term has a greater score in a given document if it appears frequently in such a document and/or it appears in few other documents. Equivalently, a term is penalized if it appears in few documents or very rarely in a document.

After applying all the steps previously described, we have a list of the most representative terms for each label. Nevertheless, many terms that appear last in the bag-of-words are not very representative, so we decided to only include the words with a TF-IDF value over 0.3. This threshold was determined because it fitted at least 75% of the words in each bag.

5.3 Decision tree using common words segment-label

We manually designed a binary decision tree by defining the steps a person would follow at the time of classifying a transcription segment. The most common interpretation of a human classification is to first identify the features or questions most discriminating that help rule out a class. Following this principle, we built the binary decision tree shown in Figure 5.2.

The tree is composed of four query nodes and five leaf nodes, each representing one teaching activity. The queries drive a segment down the tree until a leaf node is reached.

As commented above, we selected questions that discriminate among classes as soon as possible. To answer those questions we compared the vocabulary of the segment with the vocabulary of the classes involved in the query. The tree was designed by students and university lecturers with experience in lecture activity recognition even if none of them was familiar with the subjects of the recordings.

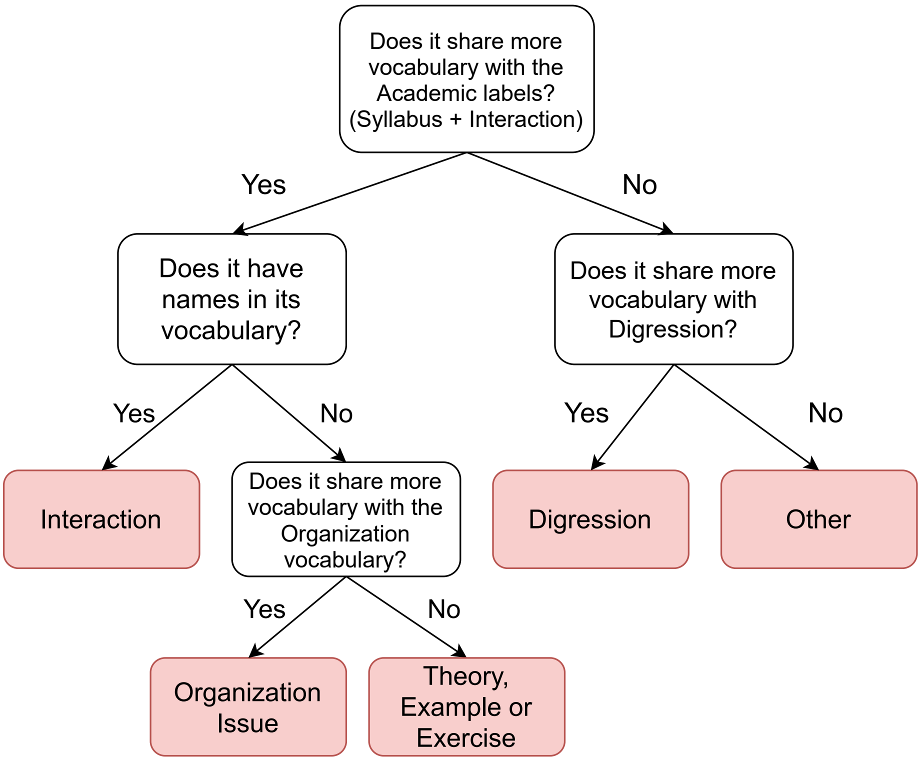

Does it share more vocabulary with the Academic labels? (Syllabus + Interaction)
Yes
No
Does it have names in its vocabulary?
Does it share more vocabulary with Digression?
Yes
No
Yes
No
Interaction
Does it share more vocabulary with the Organization vocabulary?
Digression
Other
Yes
No
Organization Issue
Theory, Example or Exercise

Figure 5.2: Classification Tree.

The principle that guided the selection of questions was that of first discarding the labels that are easier to identify on each step. Hence, firstly we split the labels that are the most different to the rest, that is, those that are academical based (the labels under *Syllabus* and *Interaction*) from those which are not (*Digression* and *Other*). Also, these two latter labels are the ones that have a lower recall value, as was shown in Table 4.4, and are thereby easier to identify.

Under the non-academic branch, we simply compared each of the vocabularies and classified under the greatest share among them. On the opposite branch, we split between *Interaction* and the rest of activities. Even though *Interaction* shares a great deal of vocabulary with the *Theory, Example or Exercise* label, interactions typically include names when lecturers refer to students, so we included this as a criterion. Moreover, if the answer to such question is negative, the last decision is between *Organizational Issues* and *Theory, Example or Exercise* labels. We created a custom vocabulary for the *Organization Issues* class in addition to its own vocabulary. This is because many words that determine if a segment is part of this label are given by this vocabulary, which includes names of weekdays, names of months and many more, you may see a part of it in Figure 5.3.

['lunes', 'martes', ... , 'noviembre', 'diciembre', 'examen', 'semana', 'día', 'mes', 'año', 'curso', 'practica', 'examen', 'horario', 'semana', 'mes', 'año', 'curso', 'examen', 'ejercicio', 'subir']

Figure 5.3: Custom Vocabulary for the *Organization* label.

As for the *Interaction* label vocabulary we created a list of the 100 most common male and female names (100 male and 100 female). Figure 5.4 depicts a section of the names used in this vocabulary.

['hugo', 'lucas', 'martin', 'daniel', 'pablo', 'mateo', 'alejandro', 'leo', 'alvaro','manuel', 'lucia', 'sofia', 'martina', 'maria', 'paula', 'julia', 'emma', 'valeria', 'daniela','alba', 'adrian', ..., 'marco', 'javier', 'marcos', 'izan', 'antonio', 'alex', 'miguel', 'carlos', 'juan', 'gonzalo']

Figure 5.4: Vocabulary for the *Interaction* label.

The results we obtained when using the tree in Figure 5.2 to classify transcription segments are shown in Figure 5.5. An initial observation can be made: in a similar fashion as with the NN model, *Organization*, *Interaction*, *Digression* and *Other* share common vocabulary with *Theory, Example or Exercise*, resulting several segments in these classes being classified in the common bulk of the academic discourse.

The overall accuracy of this experiment is an acceptable value of 60.3% (see caption label of Figure 5.5). In this experiment, we compared the segment's vocabulary to the label's vocabulary word by word, completely ignoring the context of the rest of the sentence or phrase. We then decided to go further and attempt to improve this result. The new idea lies in using pairs of consecutive words (we will call them *bi-terms*) instead of single words. We compared the bi-terms of the segments to the bi-terms of the activities and retrieved the confusion matrix shown in Figure 5.6. We can observe the outcome is much higher, obtaining a 92.7% accuracy.

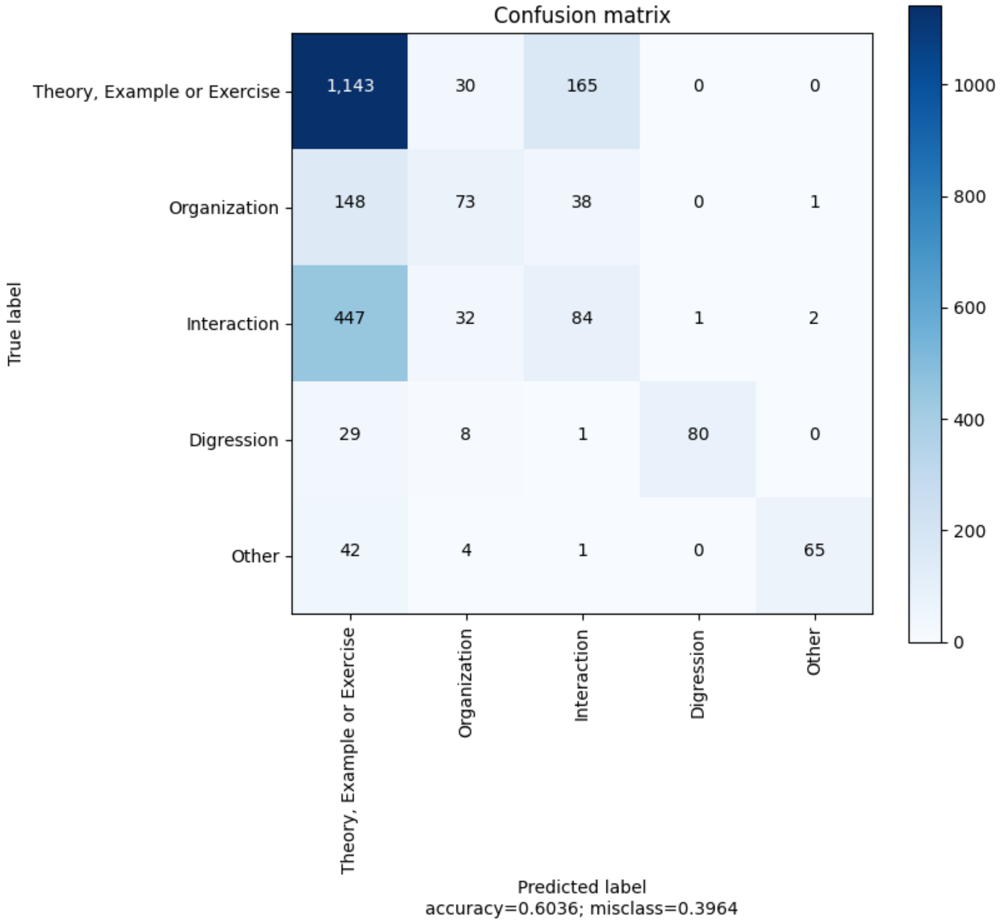


Figure 5.5: Confusion Matrix of Human-like Classification.

A deeper analysis of the confusion matrix in Figure 5.6 reveals that, although the *Theory, Example, Exercise* column values are still high (low precision in the identification of transcription segments of these activities because of the many False Positives), it is drastically lower than in Figure 5.5, which is also the main reason for the overall accuracy increase. Moreover, the *Digression* and *Other* classes are nearly perfectly classified except for the False Positives of the *Theory, Example, Exercise* class. The class *Organization* still displays some noticeable confusion with *Interaction* because students tend to ask many questions when a lecturer talks about organization issues. Nevertheless, we believe this result is within an error tolerance that indicates that the tree is not "over-fitted".

Finally, we wondered if a larger term sequence would result in a higher accuracy value. We run a third experiment using three consecutive terms (*tri-term*) but the final accuracy result dropped to 89%.

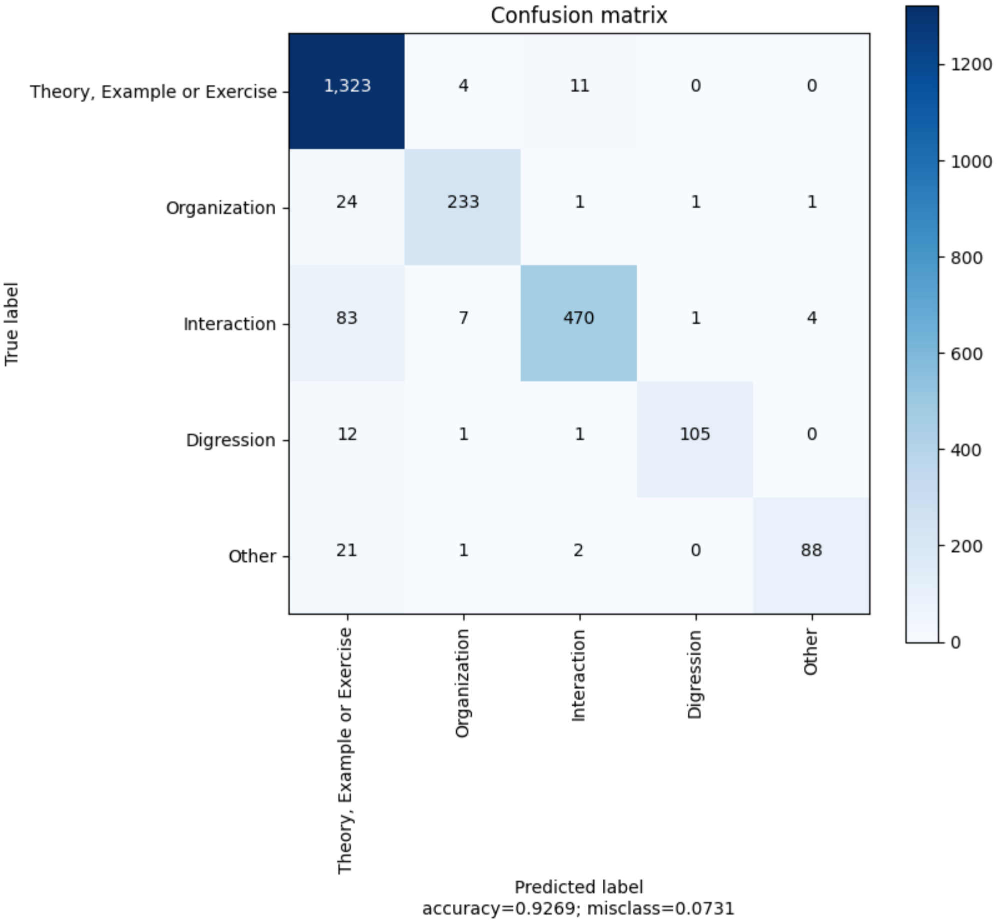


Figure 5.6: Confusion Matrix of Human-like Classification using bi-term.

5.4 Comparing the NN and Decision Tree methods

Ultimately, the aim of this chapter is to check whether a human-like classification returns better results or not that those obtained with the NN classification model.

Comparing the outcomes of the NN model in Figure 4.2 and the results of the decision tree in Figure 5.6 one can conclude that a human-like classification vastly outperforms the neural network model given that using bi-terms on the binary tree reports an accuracy of 92% vs 74% obtained with the NN model. However, this first glance may lead us to error. The neural network classifier struggles mostly when distinguishing between the *Theory*, *Example* and *Exercise* classes. Therefore, a fair comparison should consider these three labels as a single one too. Figure 5.7 shows the equivalent confusion matrix after applying the grouping correction.

As we can observe, the accuracy value increases by over 10 percentage points. Even if this is a significant improvement, vastly lower than the human-like bi-term classification. Nevertheless, with this method we obtain a higher accuracy score than with the Human-like approach using uni-terms. We obtain an only slightly lower result than when using the Human-like approach with tri-terms.

Our conclusion is that that a more human-like classification is capable of distinguishing better between *Interaction* and *Theory, Exercise and Example* activities since the largest difference between both models is caused by the misclassification error between these

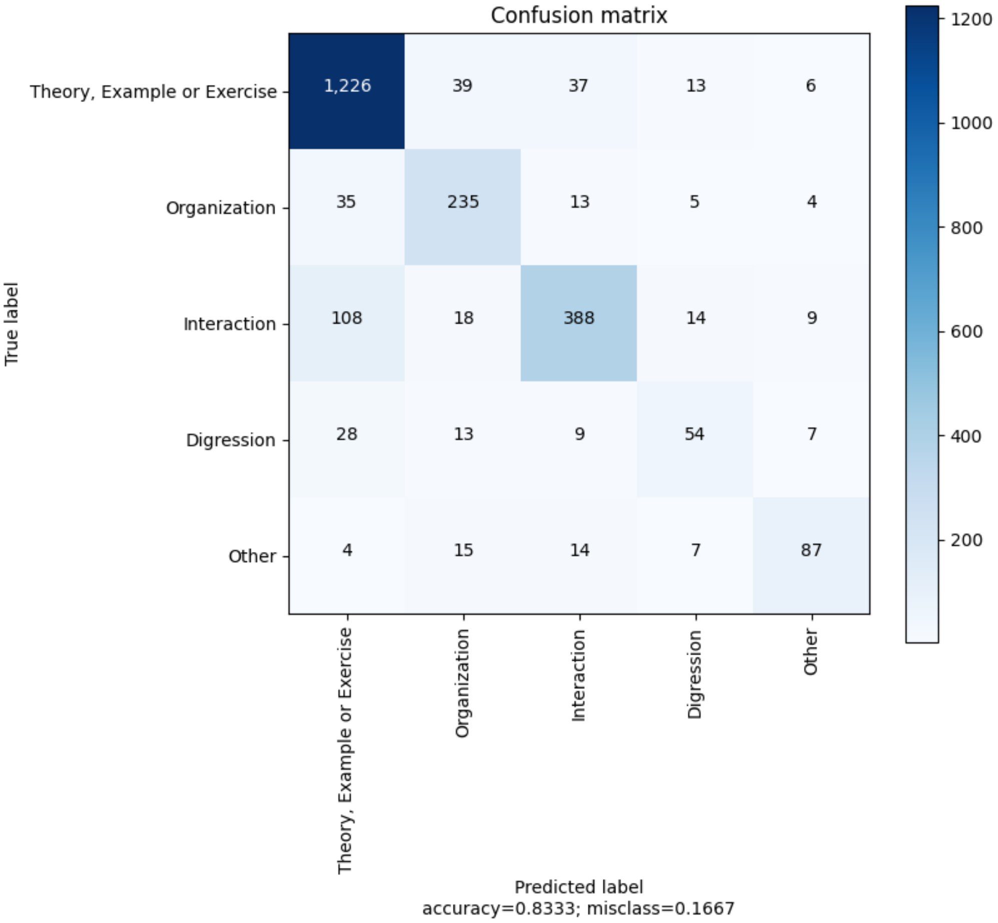


Figure 5.7: Equivalent Confusion Matrix of NN classification.

two classes. In addition, we can see that the neural network classifier is generally making more mistakes because of all the classes show some degree of misclassification, unlike the human-like classifier that has 0 or negligible values on many cells outside the main diagonal.

CHAPTER 6
Conclusions and Future Work

6.1 Conclusions

The work developed in this project has relation with many hot research topics and cross-cutting themes: corpus linguistics, genre studies theories, classification techniques, Neural Networks, Transformer models, decision trees and most importantly techniques for analyzing teaching discourse.

The main conclusions of this work are:

- We revised several works that investigate the genre of academic lecture as one of the most popular ways of instruction in universities in order to come up with the hierarchy of teaching activities presented in chapter 3. The seven final classes we worked with throughout the text were also the result of a thorough evaluation of multiple classroom audio and video recordings of the VIDEOAPUNTES repository.
- In addition, we built a recognition system for such teaching activities through automatic transcriptions of lecture recordings. Such system is built using a transformer based model that enabled us classify based on sentence content. After pondering various options, we chose *XLM-RoBERTa-large* but we came to realise it was too large for our hardware restrictions.
- Finally, we analyzed the difference between an automatic recognition system to an experienced human approach of the same classification method. We used a binary classification tree comparing the segment's vocabulary to the labels' vocabulary using bag-of-word representations to perform such task. Surprisingly, we obtained better results using the human-like classification.

We hope this technology will improve the experience of students when watching pre-recorded lectures and enable students to learn where, when they want and, more importantly, at a pace of their choosing.

6.2 Future Work

As mentioned in chapter 1, the PROMETEO project will continue to develop features for this line of research. A favorable improvement that may be added is to introduce audio and/or video recordings of the lectures to the neural network so other techniques like **pattern mining methods** and **logic-based and planning-based methods** can be applied to recognise patterns in lecturers behaviour.

As mentioned in subsection 4.3.4, one of the major improvements we can apply is to train the neural network with a larger, more balanced dataset. Although it is hard for text, it probably is advisable to do some data augmentation to increase the dataset size.

It would also be interesting to employing a Language Model to correct and spellcheck the automatic transcriptions. We believe that a source of error in our method is the existence of mistakes in transcription of the audio.

Finally, a large improvement would be to use *XLM-RoBERTa large* as it outperforms the base model. However, the tradeoff between accuracy and resource consumption (time and memory) has to be taken into account.

6.3 Applied Knowledge

This project has applied a great deal of topics related to the subjects learnt throughout the Computer Science degree. I have used topics related with courses like IIP (Introducción a la Programación y la Informática) course and PRG (Programación) course in terms of basic coding and programming. In chapter 4 I have applied knowledge learnt in the SIN (Sistemas Inteligentes) and APR (Aprendizaje Automático) courses in the training of a neural network. Lastly, in chapter 5 the SAR (Sistemas de Almacenamiento y Recuperación de Información) course has been very present in creating vocabularies and applying metrics such as the *TF-IDF*.

Bibliography

[1] Generalitat Valenciana. Consellería D'educació. Direcció General De Política Científica. Prometeo/2019/111.

[2] Inmaculada Fortanet-Gómez. Honoris Causa speeches: An approach to structure. *Discourse Studies*, 7(1):31–51, 2005.

[3] Inmaculada Fortanet-Gómez and Begoña Bellés-Fortuño. Spoken academic discourse: an approach to research on lectures. *Revista española de lingüística aplicada*, 1(8):161–178, 2005.

[4] Douglas Biber. *Dimensions of register variation: A cross-linguistic comparison*. New York: Cambridge University Press, 1995.

[5] Eniko Csomay. Academic lectures: An interface of an oral/literate continuum. *NovELTy*, 7(3):30–48, 2000.

[6] Lynne Young. *University lectures – macro-structure and micro-features*, pages 159–176. Cambridge Applied Linguistics. Cambridge University Press, 1995.

[7] Valerija Malavska. Genre of an Academic Lecture. *International Journal on Language, Literature and Culture in Education*, 3(2):56–84, 2016.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems 2017*, pages 5998–6008, 2017.

[9] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. *IEEE Comput. Intell. Mag.*, 13(3):55–75, 2018.

[10] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention, please! A critical review of neural attention models in natural language processing. *CoRR*, abs/1902.02181, 2019.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[12] A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. In *OpenAI Blog*, 2019.

[13] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019.

[16] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining, 2019.

[17] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale, 2020.

[18] UPV. VideoApuntes. https://videoapuntes.upv.es/.

[19] UPV. MLLP transcriptions. https://ttp.mllp.upv.es/index.php?page=faq.

[20] Source Forge. Transcriber. http://trans.sourceforge.net/en/presentation.php.

[21] The Audacity Team. Audacity . https://www.audacityteam.org/.

[22] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[23] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[24] T. C. Rajapakse. Simple transformers. https://github.com/ThilinaRajapakse/simpletransformers, 2019.

[25] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[26] NLTK. Natural Language Toolkit. https://www.nltk.org/index.html.