NLP, TEXT MINING AND SEMANTIC ANALYSIS

Bachelor in Data and Business Analytics BDBA SEP-2023
NLP-DBA.3.M.A

Area Information Systems and Technology

Number of sessions: 30

Academic year: 23-24

Degree course: THIRD

Number of credits: 6.0

Semester: 2º

Category: COMPULSORY

Language: English

Professor: **ALEJANDRO VACA SERRANO**

E-mail: avaca@faculty.ie.edu

Data Scientist at Instituto de Ingeniería del Conocimiento (IIC) since 2019.

Deep Learning, Computer Vision & Natural Language Processor in different institutions since 2021.

? Google Scholar profile: shorturl.at/lnp58
??Best Poster Presentation Award at the NAACL 2022 LatinxAI Workshop.
??1st prize at the SomosNLP 2022 Hackathon for the BioMedIA project: https://huggingface.co/spaces/hackathon-pln-es/BioMedIA.
??Special award for project with the highest number of Likes for BioMedIA - Hackaton SomosNLP 2022.
??1st prize in Tasks 1 & 2 (2/2) at Exist22@Iberlef22, a challenge focused on detecting sexism in texts in English and Spanish.
???Main developer of RigoBERTa, the most capable Spanish language model, reaching a new State of the Art (best model in 10/13 tasks). Link to the paper: shorturl.at/ghjlW
??Best Data Scientist Special Award - Hackaton SpainAI 2021
??1st prize @ SpainAI 2021 Hackathon Time Series challenge.
??1st prize @ Computer Vision challenge of the SpainAI Hackathon 2021.
??3rd prize @ NLP challenge of the SpainAI Hackathon 2021.
??1st Prize @ Minsait Land Classification challenge - Cajamar UniversityHack2020.

Office Hours

Office hours will be on request. Please contact at:

LinkedIn: https://www.linkedin.com/in/alejandro-vaca-serrano/

e-mail: alejandro_vaca0@hotmail.com

SUBJECT DESCRIPTION

In this course you will learn everything you need to know to start a career in NLP. We will start with the basics, understanding what NLP really is and the main fields or knowledge areas that are part of it. Then we will travel through time by learning how text tasks were solved in the previous years, and how these tools are still useful for some concrete problems. Then we will learn more complex text representation techniques, as we include semantics. This will bring us to the most recent techniques for text data mining: transformers models.

This course is designed to provide a general outlook of all the different tools that are and were used in the previous years to solve text mining problems, with a clear emphasis on the most modern and recent approach. This is different from many other NLP courses, as these techniques were developed very recently and not all companies are even aware of their existence. Luckily, I have been working with these technologies almost since they appeared, and they are my main area of expertise. You may want to check my Google Scholar Profile, where you will find some innovative papers on modern NLP. For that reason, I assure you the following program is highly updated. This is specially important, since the objective here is that you are able to solve any text problem with state of the art results when you end this course, and that is not possible without a clear understanding of the latest trends in NLP.

The methodology for the course is very practical, with many code examples for each of the theoretical topics that will be covered. Students will be mainly evaluated by their ability to use the theoretical knowledge acquired in class to develop high quality projects. This means that theory by itself will be almost worthless for this course, there is no need to memorize it, but it is very important to clearly understand it. Good coding skills in Python are a must for this course.

LEARNING OBJECTIVES

- Understand the most common tasks in NLP.
- Understand how to solve text classification and regression tasks with classical models.
- Understand the inclusion of semantics in text representation methods.
- Understand and be able to apply different Deep Learning techniques in the context of NLP.

TEACHING METHODOLOGY

IE University teaching method is defined by its collaborative, active, and applied nature. Students actively participate in the whole process to build their knowledge and sharpen their skills. Professor's main role is to lead and guide students to achieve the learning objectives of the course. This is done by engaging in a diverse range of teaching techniques and different types of learning activities such as the following:

Learning Activity	**Weighting**	**Estimated time a student should dedicate to prepare for and participate in**
Lectures	26.67 %	40.0 hours
Discussions	6.67 %	10.0 hours

Exercises in class, Asynchronous sessions, Field Work	20.0 %	30.0 hours
Group work	33.33 %	50.0 hours
Individual studying	13.33 %	20.0 hours
TOTAL	100.0 %	150.0 hours

PROGRAM

SESSION 1 (LIVE IN-PERSON)

- Presentations (15')
- Syllabus and course introduction (10')
- Evaluation format (10')
- Introduction to NLP: what it is, etc. (35')
    - Diagram presentation, what are the different tasks and approaches, a general overlook of what is possible with NLP.
    - Definitions: NLP, Comp. Ling., etc.
    - Types of data in NLP: Different types of documents, audio data…
    - Importance of NLP: amount of data, etc.
- Q&A (10')

SESSION 2 (LIVE IN-PERSON)

- Introduction to NLP: linguistics and NLP. (45')

    What is a corpus, different types of corpora.
    Existing corpora in English and Spanish.
    Annotation Guidelines.
    Analysis layers: sentence segmentation, words tokenization, lemmatization, etc.
- Practical exercise with spacy (25')
- Q&A (10')

SESSION 3 (LIVE IN-PERSON)

- Text Classification: different types. (20')

    binary classification
    multiclass classification
    multilabel classification

- What can we do with text classification? (20')

    Sentiment analysis
    topic segmentation.
    spam detection
    toxicity detection

- Bag of Words Representation of Text. (25')
- Q&A (15')

SESSION 4 (LIVE IN-PERSON)

- TFIDF (15')
- HashingVectorizer (10')
- Example of Classical ML pipeline with Python (45')
- Q&A (10')

SESSION 5 (LIVE IN-PERSON)

- Including Linguistics in the pipeline. (15')
- Example of more complete pipelines with Scikit-learn (45')
- Q&A (15')

SESSION 6 (LIVE IN-PERSON)

- What are semantics? (5')
- How do we include semantics in text representation: Word Embeddings (30')
- CBOW (15')
- Language Models (15')
- Q&A (15')

SESSION 7 (LIVE IN-PERSON)

- Word2Vec (20')
- FastText (10')
- Using Word Embeddings with Supervised Models (15')
- Practical example of how word2vec works (20')
- Q&A (15')

SESSION 8 (LIVE IN-PERSON)

- Convolutional Neural Networks for Text (35')
- Demonstration of CNN for Text (35')
- Q&A (10')

SESSION 9 (LIVE IN-PERSON)

- Recurrent Neural Networks for Text (35')
- Demonstration of RNN for Text (35')
- Q&A (10')

SESSION 10 (LIVE IN-PERSON)

- What is Attention (20')
- The Transformer Architecture (50')
- Q&A (10')

SESSION 11 (LIVE IN-PERSON)

- Pre-training of Encoder models (45')
    - BERT
    - RoBERTa
    - ALBERT
    - SpanBERT
    - DeBERTa
- Pre-training of Decoder models (25')

GPT-2
GPT-3

- Q&A (10’)

SESSION 12 (LIVE IN-PERSON)

- Pre-training of Encoder-Decoder models (25’)

BART
T5
PEGASUS

- Language Models in Spanish (45’)

BETO
BERTIN
MARIA-PROJECT
RIGOBERTA

- Q&A (10’)

SESSION 13 (LIVE IN-PERSON)

- Multilingual Language Models (20’)
- Domain Adaptation of Language Models (15’)
- Language Models for Long Sequences (35’)
- Longformer (15’)
- BigBird (10’)
- Q&A (10’)

SESSION 14 (LIVE IN-PERSON)

- Logic behind pretraining + fine-tuning approach (10’)
- NER (20’)
- Practical exercises with NER (40’)
- Q&A (10’)

SESSION 15 (LIVE IN-PERSON)

- Extractive Question Answering (20’)
- Practical Exercises on QA & Information Retrieval (50’)
- Q&A (10’)

SESSION 16 (LIVE IN-PERSON)

- Practical Exercises on QA & Information Retrieval (70’)
- Q&A (10')

SESSION 17 (LIVE IN-PERSON)

Partial Exam on the topics 1-4 (included) (1h20’)

SESSION 18 (LIVE IN-PERSON)

- Relevant Documents Retrieval (20’)
- Translation (30’)
- Practical exercises on translation (30’)

SESSION 19 (LIVE IN-PERSON)

- Summarization (30’)
- Practical exercises on automatic summarization (40’)
- Q&A (10’)

SESSION 20 (LIVE IN-PERSON)

- Generative QA (20’)
- Generative chatbots (20’)
- Time to play with the chatbot and talk to it (15’)
- Future directions (25’)
- Q&A (10’)

SESSION 21 (LIVE IN-PERSON)

- “Mapa del Expediente” project (30’)
-

Extended Q&A session on individual projects (50')

SESSION 22 (LIVE IN-PERSON)

- BioMedIA project as an example on how to do group projects (1h)
-
  Extended Q&A on group projects (20')

SESSION 23 (LIVE IN-PERSON)

- René project (1h)
- Extended Q&A on Group Projects (20')

SESSION 24 (LIVE IN-PERSON)

Individual Project Presentations

SESSION 25 (LIVE IN-PERSON)

Individual Project Presentations

SESSION 26 (LIVE IN-PERSON)

Instruction-tuning, Reinforcement Learning from Human Feedback and Strategies for Text Generation.

SESSION 27 (LIVE IN-PERSON)

Group Project Posters Presentations

SESSION 28 (LIVE IN-PERSON)

Group Projects Presentations.

SESSION 29 (LIVE IN-PERSON)

Group Projects Presentations.

SESSION 30 (LIVE IN-PERSON)

Close-up Session, highlighting the most important topics and takeaways from the course.

EVALUATION CRITERIA

For this course, evaluation will be mainly based on the individual and group projects. Theory is useless without practice in this subject, so it is required that students develop the tools needed to carry out NLP projects on their own. The final grade distribution is the following:

Criteria	Weight
Class participation	10%
Exams	10%
Individual Assignment	40%
Group Assignment	40%

INDIVIDUAL ASSIGNMENT

Students must train 3 models of different nature for a text classification task. The models must be:

Classical ML model, with or without linguistic pipeline for text preprocessing.
Static Embeddings + Recurrent or Convolutional NN, implemented in Pytorch or keras, as you wish.
Fine-tuning of a pretrained Transformers model.
The dataset students must use to carry out this assignment is Rotten Tomatoes: https://huggingface.co/datasets/rotten_tomatoes

This dataset consists of movie reviews. The dataset is balanced, with 5k positive and 5k negative reviews. Results must be given for the test set of the dataset. There is no restriction regarding the use of more datasets for training , so students can enrich the training set of rotten tomatoes with as many datasets as they want, if that enables them to obtain better results.

Students must upload to campus a zip-compressed folder called in this format: “name_surname_individual_assignment.zip”. Inside this folder, there must be 3 subfolders: one for subtask. These folders MUST be called : “classical_ml”, “static_we”, “transformers”. Inside these folders, there must be 4 files. The first file must be the python script of the final code for each part, and is always called main.py. This script has to output the test score of the model, by first fitting the model on training data. Although you will probably have to try out many different models, pipelines etc, I just want the final version for each model, not all the trials you have tried. This script has to run effortlessly, by doing python main.py in the console under the correct environment. There must also be a csv with the results, named results.csv. This csv (comma separated) has to include two columns, named: “index” and “pred”. Additionally, in the main folder (inside “name_surname_individual_assignment”), there must be a README.md explaining the different experiments that you carried out, the results that you have obtained with each approach, and a short explanation of your code, for each part of the assignment. This README should include a conclusions section. Finally, you should include a requirements.txt in each subfolder with the concrete versions of all libraries you used for that part of the assignment (expectedly you use a different environment for classical ml, static embeddings and transformers), so that by installing with `pip install -r requirements.txt` your code can be easily run.

In the virtual campus you can find an example submission file.

Rubric for individual assignment:

2 main criteria:

- Macro F1-score for the results.csv (for each model: classic, static embeddings and transformers).(40%)
- Quality of the code solution: innovativeness, creativity, technical complexity…(60%)

Criterion	Weight	1 point	2 points	3 points	4 points
Macro f1-score of classical ML pipeline	10%	Results below percentile 20 of all the class.	Results between percentile 20 and percentile 50	Results above percentile 50, below percentile 90	Results above percentile 90
Macro f1-score of Static Embeddings + RNN/CNN	10%	""	""	""	""
Macro f1-score of Transformers	20%	""	""	""	""
General code quality.	15%	Unorganized code, no comments nor documentation of the code, unclear README.	Code organized to some extent, but not much, little comments or documentation, not very clear README.	Code well organized, with in-code comments and documentation, with a clear README.	Modular code with a great organization, very clear in-code comments and documentation, explanative, clear short README.
Technical complexity of the solutions.	15%	A copy and paste of what we saw in class, with no effort from the student in trying more difficult approaches.	Based on what we saw in class, but with more technical complexity.	The technical complexity is beyond what we saw in class, showing the effort from the student in learning more than the basics. However, this complexity is at the cost of increasing code complexity.	At the same time it is complex in technical terms, but the solution is implemented elegantly, without unnecessarily increasing code complexity. Superior understanding of the student.
Innovativeness.	15%	Only models seen in class are tried.	Only variations of the models	The student tries to use some state	In all cases, the student uses state of

			seen in class are tried.	of the art solutions, carrying out research, in some cases.	the art solutions, after carrying out exhaustive research.
Communication skills	15%	Poor communication skills. Multimedia support does not help understand and follow the presentation / monotonous tone / no interest in trying to actually connect with the audience.	Some communication skills, with the use of acceptable multimedia support which helps to follow the presentation to some extent. There is an effort from the student in trying to connect with the audience. The content is clear and easy to follow.	Good communication skills, with the use of verbal and non verbal language to reach and connect with the audience. The speaker shows pride for their work and clearly explains the details of it. The multimedia support used is very useful and clear, and the content of the presentation itself is very clear.	Excellent communication skills. The speaker communicates with passion and precision. It is clear that the student has used much time not only on the project but on the presentation itself, carefully choosing the content and the multimedia support. The student shows the ability of communicating scientific or technical work as a story.

GROUP ASSIGNMENT

Students are asked to form groups of 3 students. This project requires creativity and problem-solving aptitudes, not only NLP knowledge, so students within groups must collaborate to generate innovative ideas to solve a concrete problem. The first step is that each group must design a problem they want to solve. Take BioMedIA as an example: we want a system capable of solving open questions about biomedicine in Spanish. To choose a good idea for a project, it is very important to check the data openly available for building such a system: if we don't have data to train the models we would need to build the proposed system, we should look for other viable ideas.

Then, we design all the pieces needed to produce such a system, in the case of BioMedIA this would be the passages retriever, the passages ranker, the generative question answering model… In this case, we need to know which datasets and models are openly available for each task. Then, when the whole system is designed, work can be divided among group members, so that each member trains one or more models. Students must use gradio to build an application of their solution. This application should use the models trained in the previous step. Students should upload their app to Huggingface Spaces. This is subject to changes, since Huggingface is changing its policy and there may be charges for this (until very recently this was for free). If Huggingface Spaces is suddenly unavailable, you just need to add an app.py script to your folder submission.

Additionally, there will be a "posters presentation" (here is a link about what that really is), emulating those of scientific conferences. Students will have the opportunity to design a beautiful poster summarizing their work, and they will present it to other students in a very casual way, like in real-life poster presentations. This is not the same as the final presentation students will have on their projects. For poster presentations we will prepare the classroom, and each group will have one part of the classroom as their presentation stage. At least one student per project must stay there (there will be rotation, so that all students enjoy all roles), while the others can go to the tables of other groups to listen to their poster presentations and ask them as many questions about their work as they want. In this poster presentation, students can also play around with the apps developed by other groups, so they get to really understand their apps in a practical way. This poster presentation will be graded by students from other groups, so you have to impress your classmates!

Finally, there will be a final presentation per group. Each group will have 20' to present their work. In that presentation students can give more technical details than in the poster presentation.

So, a little recap of the concrete steps for this assignment:

Choose a group. Communicate to me this group (just 1 email per group please) to my email: alejandro_vaca0@hotmail.com
Think about a problem you want to solve, or an app you would like to build (e.g.: a system that detects illnesses and drugs appearing in clinical texts and their relations and uses that information to create a knowledge graph of that). Please take into account that you need to know in advance at least some of the data that you could use for your project. Email me this problem please, just to validate your idea.
Design the app, with the datasets and models you will use for each part, in a diagram format. diagrams.io is very useful for that.
Divide the work and train the relevant models.
Create the app with gradio.
Create a poster to present, and present it.
Present your work in a more technical and detailed way.

Submission Format:

Your submission will have different parts, which are detailed below.

zip-compressed folder called in this format: “name_surname_individual_assignment.zip”, with the name and surname of one of the students in the group. This folder should contain a README.md explaining the directory structure and the different scripts contained. This README should be concise and clear, meaning that after reading it, diving through the code should not be a hassle. This folder should contain all the code used for training the models for your application, and should be uploaded to campus by April 11th, 8.00AM.
A gradio or streamlit app that should be uploaded as a Space to https://huggingface.co/ieuniversity . Students can update this app as much as they want before the due date, which is April 11th, 8.00AM. After that time no changes are allowed in the Space. A minimum requirement is that apps run without errors.
A Poster, which must be inserted into the app (in PNG format). In the BioMedIA app you have examples on how to insert an image inside an app (and there are thousands of other examples in Huggingface/spaces). This poster should be printed in A3 for Posters Presentation (April 11th, 9.00AM).
Finally you will present your projects to your classmates.
By April 12th, 9.00AM, all students should have sent an email to: alejandro_vaca0@hotmail.com with the subject “posters presentation grading”. This email should contain a grade from 1 to 10 for all the projects but theirs (self-grading is not allowed), organized in bullets, together with an explanation of that grading. This explanation should at least answer the following questions: what are the key features of the app? What are the different parts of the architecture? What impressed you the most (and the least) about the app? To what extent is the app useful, or how important is the problem solved by the app? What weaknesses and strengths do you find in the app? What have you learned during the conversation with the app’s creators?

Rules for the group projects:

Anyone talking about something not related to the group projects during the posters presentation gets a 0 on the posters’ presentation section of grading.
There cannot be any updates on the submission after April 11th, 8.00AM. After that date, you can completely focus on the final presentation, as no changes can be made in the apps or models.
If the app does not run (due to errors) on Huggingface Spaces, the whole assignment is a 0, so do not wait until the last moment to upload and test your app.

Rubric for the group assignment: TODO

criteria	**percentage**	**Learning Objectives**	**Comments**
Final Exam	0 %		
Individual presentation	0 %		
Group Presentation	0 %		
Individual work	0 %		
Group Work	0 %		
Class Participation	0 %		
Intermediate tests	0 %		
Intermediate Tests	10 %		
Individual Work	40 %		
Other	0 %		
Group Work	40 %		
Class Participation	10 %		

RE-SIT / RE-TAKE POLICY

BEHAVIOR RULES

Please, check the University's Code of Conduct here. The Program Director may provide further indications.

ATTENDANCE POLICY

Please, check the University's Attendance Policy here. The Program Director may provide further indications.

ETHICAL POLICY

Please, check the University's Ethics Code here. The Program Director may provide further indications.