Mid Thesis Report

Taxiarchis Papakostas-Ioannidis, UvA-id: 11407387

June 18, 2018

1 Personal details

My email mailto:taxiarchis.papakostas@gmail.com

My Internals supervisor email mailto:e.kanoulas@uva.nl

My Externals supervisor email mailto:g.tsatsaronis@elsevier.com

2 Main Research Question

Based on the hypothesis that in text classification of academic scientific papers, a section-based representation of text is more effective than a whole text representation, the main research question of this work can be formulated as follows:

- Can a section-based representation of text be more effective in text classification than to a whole-text representation?

The research question can be further divided into the following questions, which need to be answered first, in order to go further:

- How to represent a text, based on sections and which numerical representation is more effective?
- In which way can we predict documents class based on section classes?
- Which classification method performs better in a section-based representation of text?
- How a classifier based on sections can be trained?

3 Data-set Preparation

Elsevier has in its database more than 72 million unlabeled published articles from Scopus in an XML format. For the purpose of this research, a subset of articles is needed to be matched with their labels for constructing a final gold data-set. ArXiv [1] is an organization that contains open access published scientific articles from several domains, like Physics, Mathematics etc. Elsevier has, until now, published about 30% of the arXivs articles. Each of them is classified

with different hierarchical labels. Based on the assumption that machine learning text classification algorithms do not demonstrate a good performance when they have to distinguish lower level differences between documents, four classes from the third level of Physics domain have been chosen to test our hypothesis (figure 1).

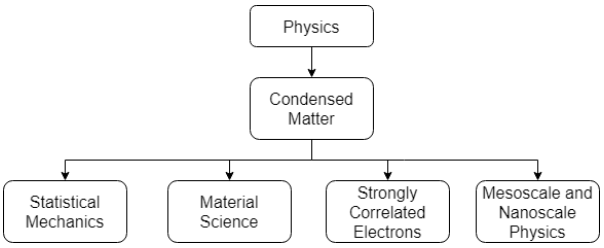

Figure 1: Task Counter

For this task, the digital object identifiers (DOIs) of the arXiv articles meta-data are needed to be matched with the Elsevier's Publisher Item Identifier (PII) for retrieving the .XML articles from the database. For achieving this, the meta-data (in .XML format) from arXiv's Physic domain articles (around 850.000) have been downloaded using its API and pursed locally by creating a list of matching DOIs per document classes (labels). After that, the DOIs have been matched with the PIIs on the Elsevier's cloud platform (Databricks) and around 7150 articles (in .XML format) have been merged with their categories based on their common DOIs. In the final stage, an Elsevier's xml Java parser has been selected among others and reconfigured for extracting the sections and subsections, in a text format, for all the matching articles. Furthermore, for the same data-set, the articles have been saved in a full document format in order to compare both representations based on the main research hypothesis question. In the end, tables have been created on Databricks by uploading the final transformed data-sets.

4 Machine Learning Pipeline - Results

For building the machine learning Pipeline, Spark and Pyspark have been used on Elsevier's Databricks [2] platform. Spark is a distributed Big Data processing environment and Pyspark is the Python API for Spark. Furthermore, spark contains higher level components like Mlib for scalable Machine learning on the cloud. The first step of the pipeline includes tokenization of all text by removing all stop words and punctuation. Moreover, all words are converted to lowercase and are represented to the vector space based on two text representation, namely tf-idf and LDA. To the extent of that, five multiclass classification algorithms are selected for this task. These are: Multinomial Naïve bayes, one-vs-all Liner SVM, Multinomial Logistic Regression, Decison Trees and Random Forest have been applied and tested on Databricks.
Therefore, two pipelines have been created, one for classifying articles as a full document and one as an ensemble classification based on the section and subsections of the articles. By performing a majority vote in the end of the

pipeline, the final class of the article is predicted. Some of the very early results are presented in tables and visually (line charts) in the appendix [6].

5 Second Half of Thesis Planning

During the second half of the thesis period, the goal is to optimize all machine learning algorithms by changing their hyper-parameters, to obtain the best results and being consistent with the evaluation. Furthermore, some statistical tests, like Fisher's exact test or a Student t-test, is needed to be applied in order to validate any significant statistical differences between our results. Finally, the plan is to experiment with different majority vote technics for synthesizing the predicted sections classes. For example, to weight differently each section on a possible experiment (i.e. Abstract, Conclusion) and apply a majority vote based on these scores.

6 Appendix

6.1 Full Document

6.1.1 TF-IDF

	Naïve Bayes	Linear SVM	Logistic Regression	Decision Trees	Random Forest
Accuracy	0.82	0.79	0.78	0.65	0.60
F1-Score	0.82	0.50	0.78	0.64	0.51
Recall	0.81	0.58	0.78	0.66	0.60
Precision	0.81	0.73	0.79	0.65	0.73

Table 1: TF-IDF Full text

6.1.2 LDA

	Naïve Bayes	Linear SVM	Logistic Regression	Decision Trees	Random Forest(150)
Accuracy	0.72	0.76	0.77	0.71	0.76
F1-Score	0.69	0.75	0.76	0.71	0.75
Recall	0.72	0.76	0.77	0.71	0.76
Precision	0.76	0.77	0.78	0.71	0.76

Table 2: LDA 20-Topics Full text

	Naïve Bayes	Linear SVM	Logistic Regression	Decision Trees	Random Forest(150)
Accuracy	-	0.77	0.73	0.72	0.76
F1-Score	-	0.76	0.70	0.71	-
Recall	-	0.76	0.73	0.72	-
Precision	-	0.77	0.76	0.72	-

Table 3: LDA 30 - Topics Full text

6.2 Per Section

6.2.1 TF-IDF

	Naïve Bayes	Linear SVM	Logistic Regression	Decision Trees	Random Forest(150)
Accuracy	0.83	0.82	0.70	0.50	0.57
F1-Score	0.83	0.81	0.66	0.44	0.45
Recall	0.77	0.76	0.67	0.49	0.55
Precision	0.77	0.76	0.71	0.71	0.71

Table 4: TF-IDF per Section

6.2.2 LDA

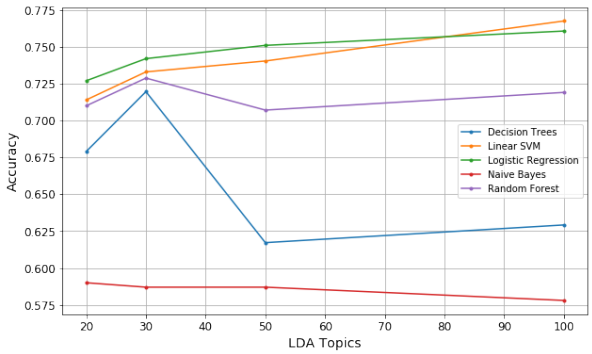

Figure 2: Accuracy based on different numbers of LDA topics

	Naïve Bayes	Linear SVM	Logistic Regression	Decision Trees	Random Forest(250)
Accuracy	0.59	0.71	0.73	0.68	0.71
F1-Score	0.46	0.66	0.69	0.64	0.66
Recall	0.57	0.68	0.68	0.63	0.67
Precision	0.56	0.68	0.67	0.62	0.66

Table 5: LDA 20 - Per Section

	Naïve Bayes	Linear SVM	Logistic Regression	Decision Trees	Random Forest(250)
Accuracy	0.59	0.73	0.74	0.72	0.73
F1-Score	0.46	0.69	0.71	0.71	0.69
Recall	0.57	0.69	0.69	0.65	0.68
Precision	0.55	0.68	0.69	0.65	0.69

Table 6: LDA 30 - Per Section

	Naïve Bayes	Linear SVM	Logistic Regression	Decision Trees	Random Forest(250)
Accuracy	0.59	0.74	0.75	0.62	0.71
F1-Score	0.46	0.71	0.73	0.56	0.67
Recall	0.57	0.70	0.71	0.58	0.67
Precision	0.72	0.70	0.70	0.51	0.68

Table 7: LDA 50 - Per Section

	Naïve Bayes	Linear SVM	Logistic Regression	Decision Trees	Random Forest(250)
Accuracy	0.58	0.77	0.76	0.63	0.72
F1-Score	0.46	0.71	0.75	0.61	0.68
Recall	0.55	0.72	0.71	0.58	0.66
Precision	0.72	0.72	0.72	0.61	0.69

Table 8: LDA 100 - Per Section

References

[1] https://arxiv.org/

[2] https://databricks.com/