NLP Course Project

Biagini Guglielmo, Castagnari Elisa, Palli Nicola, and **Terzi Angelo**
Master's Degree in Artificial Intelligence, University of Bologna
{ guglielmo.biagini, elisa.castagnari, nicola.palli, angelo.terzi2 }@studio.unibo.it

Abstract

This report outlines the use of many techniques, aimed to improve the results of several classification algorithms on the Court of Justice of the European Union (CJEU) dataset. Each algorithm deals with a task of argument mining, where the tagging involves two types of argumentative elements: premises and conclusions. Each premise can be factual, legal, or both, and each legal-type premise is associated with one or more argumentative schemes. The focus of this project is on mitigating the high imbalance between classes and improve classification in general. This is obtained by including data augmentation techniques as well as data cartography with curriculum learning. A final experiment is performed including the usage of the three of them at once. The usage of these techniques has led us to better results in some cases.

1 Introduction

We have worked on an already given dataset composed of 40 decisions from the Court of Justice of the European Union (CJEU) (Grundler et al., 2022) regarding fiscal State aid. The annotation specifies three hierarchical levels of information: argumentative elements, their types, and argumentative patterns. We have been considering three different classification tasks:

- Classify argumentative sentences as premises or conclusions.
- Distinguish between legal and factual premises.
- Identify argumentative patterns.

A significant challenge for these tasks lies in the disparity between the annotations used by NLP researchers in court decisions and how legal experts comprehend and assess legal arguments.

While computationally arguments are typically considered as arrangements of premises and claims, it is vital in the legal context to differentiate various argument types. This classification should be based on the intricate topology, legal domain, institutional position, and the cultural perspectives of authorities. In order to mitigate the unbalance of the classes we tried two different approaches. The first one is data augmentation: as done in previous work (Perçin et al., 2022), the method is based on the combination of GloVe word embeddings (Jeffrey Pennington) and the WordNet ontology (Fellbaum, 1998). In particular, there is the exploitation of WordNet to compute a set of candidate words and then choose the most similar one according to its GloVe word embedding. It has been proven that the quality of the synthetic data generated is superior to data generated by exploiting only WordNet or GloVe embeddings. It has been replaced about 60 % of the candidate terms of the original sentence.
The second approach consists of using data cartography with curriculum learning. Here the focus is on the behavior of the model on individual instances during training (training dynamics) for building data maps (Perçin et al.), revealing three distinct regions: ambiguous instances region, which contribute the most towards out-of-distribution generalization, easy-to-learn region and hard to learn; the instances that belong to the last region usually correspond to labeling errors, as demonstrated in the cited paper.
So, here there is a shift in focus from quantity to quality of data, for the purpose of obtaining more robust models and improved out-of-distribution generalization.
To apply the data cartography technique we have implemented a new neural model. This is because of the type of models used previously. These models do not have any kind of fine-tuning nor do they use techniques of optimal approximation, thus they

do not have batches and epochs during training, making it impossible to use for data cartography. Furthermore, for the same reason, their results do not depend on the order in which the samples are presented, which is the main core of curriculum learning. Indeed the curriculum learning approach is inspired by how humans learn, typically starting with simpler concepts before progressing to more complex ones. The last experiment consists of combining techniques of data augmentation, data cartography, and curriculum learning. In particular, we have selected the samples on which the augmentation is based looking at the results of the data cartography process. Indeed, the augmented data are derived from those inputs that are labeled as "hard-to-learn".

2 Background

Data augmentation is a technique used in machine learning, especially in the context of training deep neural networks, to artificially increase the size of a training dataset by applying various transformations to the existing data. The goal is to diversify the dataset, helping the model generalize better to new, unseen data and improve its robustness.
Data cartography is a method for visualizing and understanding datasets in the context of model training dynamics. It involves constructing "data maps" that provide a model-based tool to contextualize examples within a dataset. These maps are generated by leveraging training dynamics, i.e. the behavior of a model as it undergoes training. The process involves considering the mean (confidence) and standard deviation (variability) of the gold label probabilities predicted for each example across training epochs. Mathematically the metrics are defined as follows:

$$conf(x_i) = \frac{1}{E}\sum_{e=1}^{E} p_{\theta^{(e)}}(y_i^*|x_i) \quad (1)$$

$$var(x_i) = \sqrt{\frac{\sum_{e=1}^{E}(p_{\theta^{(e)}}(y_i^*|x_i) - conf(x_i))^2}{E}} \quad (2)$$

Where x_i is the ith sample, $p_{\theta^{(e)}}$ denotes the model's probability with parameters $\theta^{(e)}$ at the end of the eth epoch and E is the number of epochs.
The resulting data maps are a scatter plot of the two metrics and they reveal distinct regions within the dataset, such as instances with high variability (ambiguous for the model), instances with high confidence and low variability (easy-to-learn), and instances with low confidence and low variability (hard-to-learn). The maps can be effective tools for diagnosing large datasets, offering insights into the quality of data, and contributing to the construction of models that generalize better.
Curriculum learning is a machine learning paradigm that involves training a model on a sequence of tasks or data samples with increasing complexity or difficulty. The idea is to gradually expose the model to more challenging examples, allowing it to learn in a more structured and guided manner. This stands in contrast to traditional random sampling, where training examples are presented to the model in a random order.
The complexity of each sample is measured in terms of the results obtained by the data cartography computation: in the first place data belonging to the "easy-to-learn" region are presented, then the ambiguous ones, and finally the ones that lay in the "hard-to-learn" area.
It has been shown that this technique can improve convergence speed, final performance, and robustness of deep learning models across different domains, although results are mixed.

3 System description

For what concerns data augmentation, we used part of the code defined in the paper (Grundler et al., 2022), adapting it to our specific needs. The original code performed text classification using different embeddings (such as 'SBERT', 'LegalBERT', and 'TF-IDF') and classifiers (such as Linear SVC, K-NN,...). For each combination of the two, it performs attribute classification after a fitting phase (this, repeated five times, to test over the entire dataset) and prints the classification report. We then added the augmentation part by using part of the code found in the paper (cod), exploiting the Glove-Wordnet combined augmentation technique. This involves loading legal text data, downloading GloVe word embeddings, defining functions for synonymity and similarity operations through WordNet, and performing text preprocessing and augmentation. The goal is to generate augmented legal text data for use in natural language processing and machine learning tasks. We used this code in order to implement the following operations:

- Type Classification: a multi-label classification problem where a sentence that is known to be a premise is classified as legal (L) and/or

factual (F). All the models performed better on the majority class (factual), so we did data augmentation to increment the legal sentences.

- Argument Classification: a multi-label classification problem where given a sentence that is known to be argumentative, classify it as premise or conclusion. We did augmentation on the conclusions since they were less and premises led to better results.

- Scheme Classification: a multi-label classification task where a sentence, known to be a legal premise, is classified according to its scheme; due to the low number of samples in the dataset of Aut and Princ, we manage to get more through data augmentation on this scheme. Since this hasn't generated enough new samples what we did was change the function called 'mostsimilar' in the code where we added the 'order' argument, in this way, we added more phrases because it generates more synonyms. In the end, we will have more sentences to add to our new dataframe.
  The same has been done for the AC task since we tried to increment the order and obtained better results. This generates more phrases because, for every initial phrase you have, instead of generating just a similar one, you will generate two or more based on the order you choose. However, this value cannot be set too high, since we do not want our models to overfit the training set.

We made slight modifications in the generation of the output in order to visualize more things such as the weighted average and the sample average in this way it was easier to manage to do a comparison with the results mentioned in the paper where they did not use data augmentation.
Regarding the data cartography, firstly, as said before, we have implemented our classificator: a BERT-based model. In particular, we have selected the "prajjwal1/bert-tiny" card and used it in two ways:

- in **AutoTokenizer.from_pretrained**(): to preprocess input text data, converting it into a format suitable for input to the corresponding pre-trained model

- in **AutoModelForSequenceClassification.from_pretrained**: to load a pre-trained model for sequence classification tasks in an automatic manner.

Both methods are taken from the Transformers package contained in the HugginFace library. During the training the first layer of the BERT-based model is frozen, meanwhile the remaining part is fine-tuned on our specific tasks.
The output of the model for each input sentence is a vector containing the score for each class. That vector is then converted into a probability vector (using the soft-max function for the multi-class task and the sigmoid for the multi-label one), that is stored for each input at each epoch. Finally, those probabilities are used to compute the data cartography metrics (i.e. confidence and variability as defined before) and to plot the data map. Once we have plotted the map, we add to our dataset a new column, called "LE", that for each sample stores a metric, based on confidence and variability, that tells how much is easy for the model to classify the sample (the higher the easier). We have defined that metric for each sentence i as:

$$LE(i) = conf(i) * (1 - var(i)) \qquad (3)$$

where "LE" stands for "Learning Ease", meanwhile "conf(i)" and "var(i)" are the confidence and the variability of the model on that particular sample.
At this point, the dataset is "split" into k different datasets, each containing an increasing number of samples: the first one contains only the easy samples. Then more difficult ones are added until the last dataset corresponds to the complete one. Before starting the new training with curriculum learning we reset the parameters of the BERT-based model, and then we perform a variable (decreasing) number of epochs for each reduced dataset.

4 Data

The Demosthenes dataset (cod) that we have been worked with has the following structure: the dataset is constructed from a source corpus consisting of 40 decisions on fiscal State aids by the Court of Justice of the European Union (CJEU). These decisions, written in English, span from 2000 to 2018. The annotation scheme involved three hierarchical levels: argumentative elements (premises and conclusions), types of premises (legal and/or factual), and argumentation schemes. The elements were marked with unique identifiers, and premises were further classified. Six argument schemes were

identified: Rule (established rule), Precedent (past case precedent), Authoritative (indication by an authority), Classification (classification of a concept), Interpretative (interpretation of a legal source), and Principle (application of a general legal principle). Premises could be assigned multiple schemes if relevant.
In particular for data augmentation we have considered these parts of the dataset:

- Argument Classification (AC): we have selected the sentences known to be argumentative and that were classified as conclusions.
- Type Classification (TC): we have selected the sentences known to be a premise.
- Scheme Classification (SC): we have selected legal premises.

Finally, we have split the dataset into two parts, a training set, and a test set. The proportion of the dataset included in the test split is 25%.

5 Experimental setup and results

We utilized the Data Augmentation pipeline implemented in (cod) to compare the obtained results with those achieved without Data Augmentation. The used ML models are the following:

- A linear SVC;
- A Gaussian Naive Bayes;
- A Random Forest Classifier;
- A K-NN Classifier;
- A Polynomial SVC.

We opted to utilize the default hyperparameters without conducting a Grid Search, allowing for a direct comparison of results without any alterations to the classifiers. The main functions presented in (cod) to apply Data Augmentation have been slightly modified, in order to produce a higher number of new samples. The main working idea remains the same: new samples are generated starting from the original ones, by replacing some words with their synonyms. For the neural model used in data cartography, we have employed the "AdamW" optimizer in the training process, leveraging its advantages in adaptive learning rate adjustment and weight decay. The adaptive learning rate feature promotes faster convergence and enhanced performance. Simultaneously, the weight decay aspect, involving the addition of an L2 penalty on the weights, serves as a regularization technique to mitigate overfitting during training.
We used Binary Cross-Entropy as the metric, measuring the dissimilarity between the target and the predicted probabilities. This loss function is well-suited for multi-label classification problems, as it can be decomposed into multiple binary classifications, since instances may have multiple labels simultaneously.
It is noted that BCE is not the best choice for the AC task (being it part of the multiclass category of problems). However, knowing that the difference with Categorical CE is considered minimal and the results are already satisfying in terms of F1-score, we decided to stick with it to improve code similarity across tasks.
To assess the performance of the model on each task, we have used the Macro F1 score, comparing the results obtained with and without curriculum learning (with and without data augmentation). In the following tables, we reported some relevant results. All the results shown were obtained on the test set.

Embedding	**Classifier**	F1	F1 (DA)
TF-IDF	Linear SVC	0.87	**0.96**
	Random Forest	0.88	0.96
	Gaussian NB	0.84	0.95
	K-NN	0.81	0.90
	SVC	0.82	0.95
SBERT	Linear SVC	0.85	0.94
	Random Forest	0.86	0.94
	Gaussian NB	0.81	0.86
	K-NN	0.84	0.92
	SVC	0.87	0.94
LegalBERT	Linear SVC	0.80	0.93
	Random Forest	0.86	0.95
	Gaussian NB	0.86	0.89
	K-NN	0.88	0.93
	SVC	0.85	0.95

Table 1: *AC task with/without Augmentation.*

6 Discussion

As evidenced by Table 1, the results for the first task have shown a significant improvement due to the data augmentation technique. This improvement is

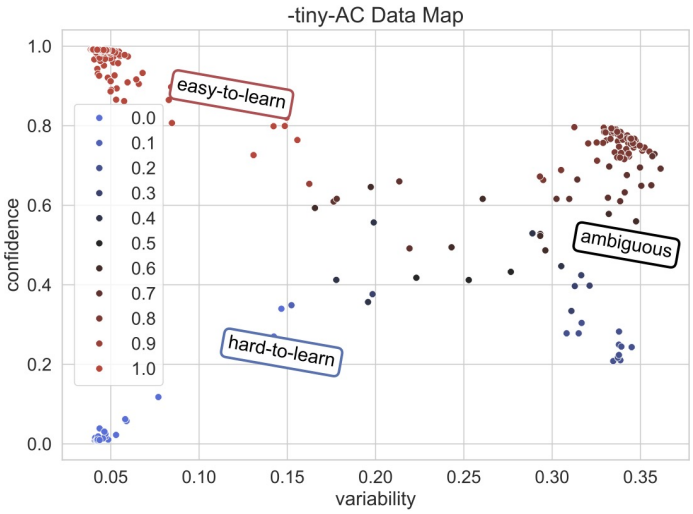

Figure 1: Cartography AC

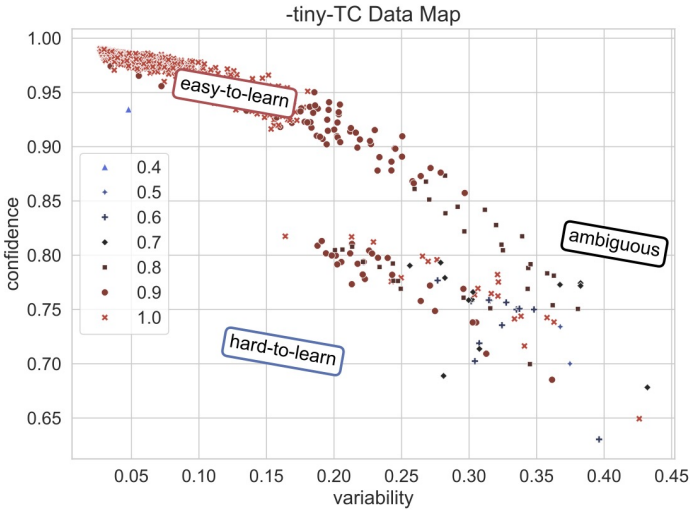

Figure 2: Cartography TC

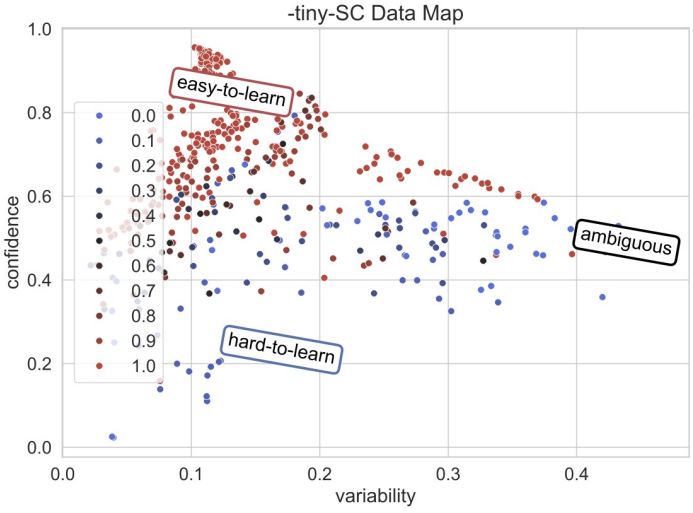

Figure 3: Cartography SC

Embedding	Classifier	F1	F1 (DA)
TF-IDF	Linear SVC	0.83	0.89
	Random Forest	0.82	0.90
	Gaussian NB	0.68	0.80
	K-NN	0.76	0.83
	SVC	0.61	**0.93**
SBERT	Linear SVC	0.77	0.82
	Random Forest	0.74	0.82
	Gaussian NB	0.72	0.73
	K-NN	0.72	0.76
	SVC	0.80	0.85
LegalBERT	Linear SVC	0.81	0.86
	Random Forest	0.77	0.86
	Gaussian NB	0.73	0.75
	K-NN	0.78	0.84
	SVC	0.85	0.88

Table 2: *TC task with/without Augmentation.*

Embedding	Classifier	F1	F1 (DA)
TF-IDF	Linear SVC	0.75	**0.83**
	Random Forest	0.73	0.76
	Gaussian NB	0.44	0.64
	K-NN	0.60	0.65
	SVC	0.31	0.48
SBERT	Linear SVC	0.66	0.75
	Random Forest	0.46	0.46
	Gaussian NB	0.54	0.50
	K-NN	0.47	0.47
	SVC	0.51	0.53
LegalBERT	Linear SVC	0.74	0.78
	Random Forest	0.51	0.48
	Gaussian NB	0.64	0.60
	K-NN	0.53	0.56
	SVC	0.64	0.59

Table 3: *SC task with/without Augmentation.*

Task	F1	F1(CL)	F1(DA)	F1(DA+CL)
AC	0.88	0.88	0.87	0.88
TC	0.86	0.85	0.85	0.85
SC	0.41	0.41	0.41	0.41

Table 4: *Results of our model*

also reflected in the TC task, as indicated in Table 2. In each case, the F1 macro score of every classifier increased by at least 0.05. Notably, the SVC with TF-IDF experienced the most substantial improvement, with a delta of 0.32, surpassing expectations. This enhanced improvement may be attributed to the binary nature of the task, while for the multi-label class imbalance posed significant challenges, resulting in a decrease in the macro score in some cases. This observation is further supported by the SC task analysis, where numerous labels are involved (Table 3). While improvements are evident across the board, some instances exhibit slightly lower F1 macro scores, such as those featuring Legal-BERT Embedding and the Random Forest classifier, which experienced a minor decline.

We are taking into consideration the test set. We have chosen to present only the macro-F1 results in the table. However, upon examining each label, it becomes apparent that augmented labels consistently demonstrate improvement. For instance, in the AC task, the F1 score associated with the "conclusions" label increased from approximately 0.7 in the unaugmented case to around 0.8 in the augmented scenario. Similar improvements were observed in the TC task, where the F1 score for the "legal" augmented case rose from approximately 0.7 to 0.8, and in the SC task, where the "AUT" augmented case saw an increase from around 0.4 to 0.5. So we reached in general better results, that was what we were expecting.

Our model showed comparable results, with respect to the predefined classifiers, in all three tasks, even if the f1-macro score of the Scheme Classification is slightly below average.

For what concerns data cartography it is important to note that we have decided to use a very simple BERT-based model (i.e. tinybert) because using bigger architecture led us to very good results, making it impossible to show changes or improvements with curriculum learning and/or data augmentation. Looking at the table 4 we can see that in the first two cases (AC and TC) curriculum learning does not improve the performance of the model. This can be attributed to the fact the model was already having great results on those tasks. Indeed, if we look at figures 1 and 2 we can see that the greater part of the samples are in the easy-to-learn region, so the main purpose of the curriculum learning technique (start the training with easy instances and increase the difficulty step-by-step) fails, since there are not enough hard-to-learn samples. However, we cannot notice improvements even in the Scheme Classification task where, as shown in figure 3, there is a more balanced distribution between the three regions. This result was not expected but probably is related to the lack of data for such a complex task.

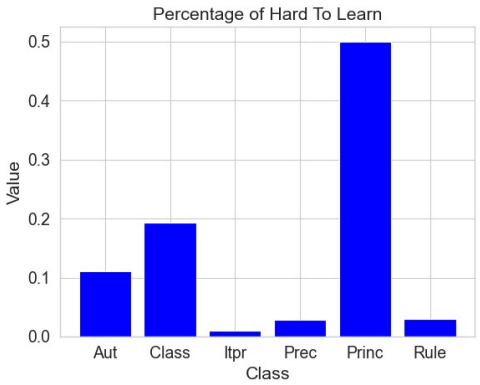

Figure 4: Scheme Classification

For this reason, we also try to use the data augmentation technique applied before, but with a slight difference: in this case, the samples to augment are chosen within the hard-to-learn sentences. To understand why we have done that, we need to focus on the hard-to-learn samples for each task: in all three cases, the instances belonging to the hard-to-learn region of the data map are the ones that are part of the less represented class. For example, in the last task (Scheme Classification) the percentage of hard-to-learn samples belonging to the "princ" class is more than ten times higher than the one of the most represented classes ("itpr", "prec" and "rule"). This is due to the fact that "princ" has very few samples. For this reason, the category has not been considered in the original paper. Accordingly, as we can see in figure 4, also the percentage of hard-to-learn authoritative and classification schemes is way higher with respect to the classes with more samples.

Still, also in this case the performance of our model did not improve.

Finally, we tried to train the model with curriculum learning on the augmented dataset, but once again we did not get any better results.

This did not match our expectations, but it can be due to several factors. First, even using a very simple BERT-based model, we obtained very good results on the first two tasks, so it would have been very difficult to improve them. Indeed, good results lead to a very low number of hard-to-learn samples and, as a consequence, a very low number of augmented instances. Then, it is proven (Soviany et al.) that in some cases curriculum learning does not improve the model, but actually makes it worse leading the model to a suboptimal solution. This occurs due to the presence of additional elements that affect performance, and these elements may suffer adverse effects from curriculum learning approaches. For instance, if the difficulty metric tends to favor selecting simple instances from a limited set of classes, it reduces the variety of data samples during the initial training phases.

7 Conclusion

We have presented several techniques, including data augmentation, curriculum learning, and a combination of the two, aimed at enhancing the performance of various classifiers, including a neural network developed by us. The data augmentation technique we adopted has shown satisfactory performance across all three tasks and all predefined classifiers, with particularly notable improvements observed in the Scheme Classification task. Overall, our approach highlights the significant negative impact of data imbalance on the results.

However, our model did not yield the same favorable outcomes, neither with curriculum learning nor with data augmentation.

A potential solution could involve gathering more authentic data to include more challenging samples, thereby enhancing the effectiveness of curriculum training. Alternatively, one could explore variants of this method available in the literature, such as "Self-Paced Learning" (Soviany et al.), where the sample order is dynamically adjusted based on the model's performance, or "Balanced Curriculum" (Soviany et al.), which incorporates multiple ordering criteria. Moreover, curriculum learning could be expanded to ensure the representation of all classes in the intermediate datasets. However, these options require further investigation.

8 Links to external resources

Dataset:
https://github.com/adele-project/demosthenes

References

Adele Project Demosthenes.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

Giulia Grundler, Piera Santin, Andrea Galassi, Federico Galli, Francesco Godano, Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, and Paolo Torroni. 2022. Detecting arguments in CJEU decisions on fiscal state aid. In *Proceedings of the 9th Workshop on Argument Mining*, pages 143–157, Online and in Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Christopher D. Manning Jeffrey Pennington, Richard Socher. Glove: Global vectors for word representation.

Roy Schwartz‡ Perçin, Swabha Swayamdipta†, Yizhong Wang Nicholas Lourie†, Noah A. Smith† Hannaneh Hajishirzi†, and Yejin Choi†. Dataset cartography: Mapping and diagnosing datasets with training dynamics.

Sezen Perçin, Andrea Galassi, Francesca Lagioia, Federico Ruggeri, Piera Santin, Giovanni Sartor, and Paolo Torroni. 2022. Combining WordNet and word embeddings in data augmentation for legal texts. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 47–52, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. page 34.