Legal Text Augmentation: Enhancing Datasets for Classification Tasks with WordNet+GloVe and Agent-Based Methods

3-cfu Project Work

Master's Degree in Artificial Intelligence, University of Bologna
{ safoura.banihashemi, alexcristian.cozma }@studio.unibo.it

Abstract

Enhancing low-frequency categories in a legal text classification dataset remains a significant challenge. To address this problem, we explore two data augmentation strategies: one based on the combination of WordNet and GloVe embeddings and the other applying agent-based generation with large language models (LLMs). We apply these methods on the Scheme (Class, Aut, Princ) and Name (Conc) categories in multi-label dataset of legal decisions from the Court of Justice of the European Union.
Each original sentence receives a single augmented version to enrich the target classes for a downstream classification task across the Scheme, Name, and Type categories.
A descriptive comparison using ChatGPT-4.1 indicates that the agent-based augmentation method produces reliable and legally appropriate augmented texts. However, the classification results show that the WordNet + GloVe (WN+GV) method achieves higher F1 scores for both the Scheme and the Name categories, suggesting that greater data diversity introduced by WN+GV.

1 Introduction

Data augmentation is a crucial strategy in natural language processing (NLP) to expand the diversity and quantity of data, especially in domains where labeled datasets are small, imbalanced, or sparse in key categories. In the legal domain, augmentation is challenging and essential due to the formal language, complex structure, and low-resource nature of legal texts.
This report introduces two augmentation methods: a combination of WordNet and GloVe (WN+GV), and agent-based augmentation using large language models (LLMs). These techniques are applied to create a more balanced dataset for classification tasks across the Scheme, Name, and Type categories.

2 Background

Classification tasks in NLP are highly dependent on the quality and quantity of the dataset. In legal text classification, label imbalance is a main challenge, as some legal categories have rare sources, and traditional augmentation methods, such as synonym replacement, have limited effectiveness in this context due to the formal and strict syntactic requirements of legal language.
Recent research on enhancing legal text using WordNet combined with GloVe embeddings has shown improved diversity while preserving legal semantics. (1) In this project, we introduce an agent-based augmentation approach using LLM and compare its performance with the WordNet+GloVe method in classification tasks. In addition, we perform a qualitative analysis using ChatGPT-4.1 to descriptively compare the outputs of both augmentation methods. Although an expert legal opinion is necessary, this approach provides valuable insight into the differences between the two augmentation strategies.

3 System description

Before augmentation, the dataset was cleaned to reduce noise. A cleaning function was implemented with the following steps:

- Lowercased all text
- Removed HTML tags such as `<ref>`
- Removed all numeric characters
- Identified and replaced 38 unusual characters (e.g., `‘, ’, ñ, û, >, ´`)
- Removed punctuation characters
- Normalized whitespace into single spaces

This process ensured a uniform input format for both the augmentation models and the downstream

tasks.
The two augmentation strategies were developed as follows:

3.1 WordNet + GloVe Augmentation

This augmentation method replaced selected words with semantically similar synonyms from WordNet, ranked by cosine similarity using GloVe embeddings. The method aimed to replace 60% eligible words, defined as adjectives, nouns, and adverbs, excluding stopwords.
This approach balances lexical diversity with semantic preservation, allowing controlled transformations while minimizing distortion of legal meaning.

3.2 Agent-Based LLM Augmentation

This augmentation method used a single-agent LLM-based was deployed via the Camel framework. The agent using the Google Gemini 2.5 Flash model via OpenRouter, guided by the following system prompt:

> You are a text generation agent. You will receive legal text as context, and your objective is to analyze this context and generate an augmented version of the text that preserves its original meaning. The output should be of similar length to the input.

The classification pipeline is organised as follows:

- **Offline augmentation.** Augmentation is performed outside the classification notebook; the two augmented corpora (`WN+GloVe` and `WN+POS`) are then loaded directly from Google Drive.
- **Normalisation.** Ensuring that every corpus has a list in `"Scheme"` and a string in `"Name"` to prevent further failure during the evaluation run
- **Utility definition & model initialisation.**
  - *Embeddings:* TF–IDF, SBERT (`bert-base-nli`), Legal-BERT-Small.
  - *Classifiers:* Linear SVC, Random Forest, Gaussian NB, k-NN, polynomial SVC, plus random and majority baselines.
  - *Tasks:* AC, TC, `SC_All`, `SC_-Princ`. Here `split`=1 is the validation set, `split`=2 is the test set, and `split`=3–5 form the training set; these indices are provided directly by the corpus.
  - Coding: General inspiration was taken from the Demosthenes GitHub.
- **Evaluation.** The notebook prints tables of macro–F_1 scores for each augmentation method and baseline, also F1 scores for each class as done in the original article in the demosthenes github. A comparison of the scores is provided, at the end, focusing on the Scheme classification, the improvement associated with the augmentation is shown for each class.

4 Data

The Demosthenes corpus consists of English-language text segments extracted from decisions of the Court of Justice of the European Union (CJEU) regarding fiscal state aid. It contains a total of 2,535 annotated legal segments.

The class distribution across both the Scheme and Name categories is unbalanced. Additionally, the Scheme category presents a multi-label classification challenge. To avoid inadvertently increasing the number of samples in non-target classes, we applied augmentation only to those samples that were labeled with a combination of the three target classes: Princ, Aut, and Class.

The results of applying augmentation are presented below:

- **Scheme Category (multi-label)**:
  - `Princ`: Increased from 16 to 20 samples
  - `Aut`: Increased from 53 to 86 samples
  - `Class`: Increased from 56 to 83 samples
- **Name Category (single-label)**:
  - `Conc`: All 160 original samples were augmented, resulting in 320 samples
- **Type Category**:
  - `L`: 906 samples
  - `F`: 1,576 samples
  - No augmentation was applied, as this category was already balanced

To preserve label integrity during augmentation, we set non-related categories to `None` for each augmented sentence. For example, when augmenting a Scheme sample, we set the Name and Type to `None`.

Each augmentation method produced one augmented sentence per original sample in the selected classes. In total, 224 new segmented sentences were added to the dataset.

5 Experimental setup and results

We are applying a descriptive evaluation using ChatGPT-4.1. Although we were aware that this approach is not reliable and expert opinions are necessary, our aim was to confirm the acceptability of the augmentation with the agent and to examine the differences between two augmentation methods. The WN+GV method attempts to augment sentences while preserving the main structure, whereas the agent-based method introduces new structures, such as converting positive sentences to negative ones and vice versa. Although ChatGPT-4.1 suggests that agent-based augmentation produces more reliable and contextually appropriate sentences for legal texts than WN+GV, we do not consider this claim reliable without further validation. To gain insight into the differences between these two augmentation methods, we present some samples below and score them as follows: +1 for good changes, 0 for neutral changes (neither good nor bad), and -1 for errors.

Example 1:

original sentence: second the intervention must be liable to affect trade between member states.

WN+GV method: second base the treatment must be liable to impact trade in between member states.

Agent-base method: Secondly, the intervention must be likely to impact trade among member states.

Table 1 shows that WN+GV introduces two errors, while the agent-based augmented sentence is more correct and fluent.

Table 1: word replacements based on the context

Original	WN+GV	score	Agent	score
second	second base	-1	secondly	+1
intervention	treatment	-1	-	-
between	-	-	among	0
affect	impact	+1	impact	+1

Example 2:

original sentence: accordingly the first limb of the first plea cannot succeed and must be rejected.

WN+GV method: consequently the for the first time arm of the first plea cannot succeed and must be rejected.

Agent-base method: Consequently, the initial component of the primary argument fails and should be dismissed.

Table 2 shows that while both methods have some problems, WN+GV handles this sentence better in augmentation.

Table 2: word replacements based on the context

Original	WN+GV	score	Agent	score
accordingly	consequently	+1	consequently	+1
first limb	for the first time arm	-1	initial component	0
first plea	-	-	primary argument	0
cannot succeed	-	-	fails	+1
must be rejected	-	-	should be dismissed	-1

Example 3:

original sentence: therefore the commission's second plea must be rejected as unfounded.

WN+GV method: thus the military commission's second base plea must be rejected as unfounded.

Agent-base method: Therefore, the Commission's second plea cannot be accepted as valid.

Table 3 shows that WN+GV changes the meaning of the sentence by adding "military commis-

sion," while the agent method produces a better augmentation, though it uses softer language.

Table 3: word replacements based on the context

Original	WN+GV	score	Agent	score
therefore	thus	+1	-	-
commission	military commission	-1	-	-
second	second base	-1	-	-
must be rejected as unfounded	-	-	cannot be accepted as valid	0

Note: These results were obtained by evaluating both grammar and meaning using ChatGPT-4.1.

The classification tasks use the original 5-fold split:

- train = {Split = 3, 4, 5}
- val = {Split = 1}
- test = {Split = 2}

Input representations

- TF–IDF (scikit-learn defaults)
- SBERT `bert-base-nli-mean-tokens` (frozen weights)
- Legal-BERT-Small `nlpaueb/legal-bert-small-uncased` (frozen)

Classifiers Linear SVC, Random Forest, Gaussian NB, k-NN, polynomial-kernel SVC, plus random and majority baselines. Multi-label tasks wrap real classifiers in `OneVsRestClassifier`.

Hyper-parameter grids

Model	Grid explored on the validation fold
Linear SVC	$C \in \{0.1, 1, 10\}$
Random Forest	$n_{\text{estimators}} \in \{100, 300\}$
SVC (poly)	$C \in \{0.1, 1\}$, degree $\in \{2, 3\}$
k-NN	$k \in \{3, 5, 7\}$
Gaussian NB	(no tunable parameters)

For each embedding–classifier pair, every grid point is trained on the train fold and evaluated on the validation fold. The best setting is re-trained on train+val and finally tested on the held-out test fold.

Metric Macro-averaged F_1 is reported for every task. For the multi-label `SC_All` task we also show (i) raw label counts (Table 12) and (ii) the mean F_1 for every scheme label (Table 13). Tables 4–11 list, *separately for each augmentation*, the five largest positive macro-F_1 deltas (augmentation – baseline).

Table 4: Top-5 (Aug-GV − Base) gains — **Argument Component (AC)**

Embedding	Classifier	Δ_{GV}
tfidf	Linear SVC	+0.076
sbert	SVC (poly)	+0.063
tfidf	k-NN	+0.062
tfidf	Gaussian NB	+0.058
legal	k-NN	+0.057

Table 5: Top-5 (Aug-GV − Base) gains — **Type Classification (TC)**

Embedding	Classifier	Δ_{GV}
legal	Random Forest	+0.013
tfidf	Random Forest	+0.012
tfidf	Random	+0.009
legal	Random	+0.008
sbert	Random	+0.005

Table 6: Top-5 (Aug-GV − Base) gains — $\mathbf{SC_{All}}$

Embedding	Classifier	Δ_{GV}
sbert	Linear SVC	+0.239
tfidf	Random Forest	+0.177
legal	k-NN	+0.112
legal	SVC (poly)	+0.083
tfidf	k-NN	+0.077

Table 10: Top-5 (Aug-Agent − Base) gains — $\mathbf{SC_{All}}$

Embedding	Classifier	Δ_{Agent}
tfidf	Random Forest	+0.190
tfidf	k-NN	+0.083
tfidf	SVC (poly)	+0.056
sbert	Linear SVC	+0.043
sbert	k-NN	+0.040

Table 7: Top-5 (Aug-GV − Base) gains — **SC$_{\text{-Princ}}$**

Embedding	Classifier	Δ_{GV}
tfidf	Random Forest	+0.171
legal	k-NN	+0.134
legal	SVC (poly)	+0.101
sbert	Linear SVC	+0.097
tfidf	k-NN	+0.087

Table 8: Top-5 (Aug-Agent−Base) gains — **Argument Component (AC)**

Embedding	Classifier	Δ_{Agent}
tfidf	Linear SVC	+0.069
sbert	SVC (poly)	+0.052
tfidf	k-NN	+0.049
legal	k-NN	+0.042
tfidf	Gaussian NB	+0.042

Table 11: Top-5 (Aug-Agent − Base) gains — **SC$_{\text{-Princ}}$**

Embedding	Classifier	Δ_{Agent}
tfidf	Random Forest	+0.194
tfidf	k-NN	+0.114
tfidf	SVC (poly)	+0.067
sbert	Linear SVC	+0.042
sbert	k-NN	+0.037

Table 12: Label counts in **SC$_{\text{All}}$**

Scheme	Baseline	Aug-GV	Aug-Agent
Prec	504	504	504
Rule	323	323	323
Itpr	298	298	298
Aut	53	86	86
Class	56	83	83
Princ	16	20	20

Table 13: Mean F_1 per scheme — **SC$_{\text{All}}$** (all models averaged)

Scheme	Baseline	Aug-GV	Aug-Agent	Δ_{GV}	Δ_{Agent}
Aut	0.080	0.233	0.173	+0.152	+0.093
Class	0.547	0.580	0.568	+0.033	+0.020
Itpr	0.133	0.131	0.131	–0.002	–0.002
Prec	0.555	0.552	0.551	–0.003	–0.004
Princ	0.102	0.148	0.101	+0.046	–0.001
Rule	0.384	0.378	0.376	–0.006	–0.008

Table 9: Top-5 (Aug-Agent − Base) gains — **Type Classification (TC)**

Embedding	Classifier	Δ_{Agent}
legal	Random	+0.026
tfidf	Random Forest	+0.016
sbert	Random	+0.018
legal	Random Forest	+0.012
tfidf	Linear SVC	+0.000

Table 14: Comparison of *conc* and *prem* Across Settings

	Baseline	Aug-GV	Aug-Agent	Δ GV	Δ Agent
conc	0.561	0.629	0.612	0.067	0.051
prem	0.939	0.924	0.923	- 0.015	-0.016

Δ-summary (mean | median macro-F_1)

Task	WordNet+GloVe	Agent
AC	0.0263 | 0.0263	0.0176 | 0.0166
TC	0.0022 | 0.0000	0.0023 | 0.0000
SC$_{\text{All}}$	0.0368 | 0.0048	0.0164 | 0.0000
SC$_{\text{-Princ}}$	0.0341 | 0.0127	0.0206 | 0.0099

6 Discussion

The augmentation strategies evaluated in this project indicates that the WN+GV method reliably improved macro-F_1 scores, across both Scheme and Name categories for low-frequency classes, Aut, Princ, and Class in Scheme and Conc in Name. In contrast, the agent-based LLM approach produced more diverse and creative sentence structures, which may help models generalize but also risk of minor shifts in meaning from the original meaning.

Importantly, careful control was applied to avoid accidental data increase in non-target categories, preserving dataset integrity and addressing the multi-label challenge in Scheme both indicate the value of targeted approaches for complex legal tasks, but expert review remains crucial to ensure legal validity.

7 Conclusion

This work indicates the effectiveness of combining WordNet with GloVe embedding and LLM-based agent augmentation strategies to address class imbalance in legal text datasets. Although the WN+GV method achieves stronger macro-F_1, across both the Scheme and the Name categories for the target classes, the agent-based method,

by introducing new structures and more creative augmentations, underscores the importance of qualitative evaluation and expert review in future work.

8 Links to external resources

- The Demosthenes corpus used in this project can be accessed at the following GitHub repository:
  https://github.com/adele-project/demosthenes

References

[1] Combining WordNet and Glove Embeddings. https://aclanthology.org/2022.nllp-1.4/

[2] Camel AI framework. https://github.com/camel-ai/camel

[3] ChatGPT by OpenAI was used as an assistive tool for clarifying concepts and refining the work.