Finding the root in free undirected syntactic trees

Gabriel Guerra, Adrià Espinoza

I. Introduction

This project aims to identify the root node in undirected syntactic dependency trees provided across 21 languages. Each sentence is represented as a graph, where words are nodes and undirected syntactic dependencies are edges. The problem is framed as a binary classification task at the node level: predicting whether a node is the root. The primary goal is to maximize sentence-level accuracy (i.e., selecting the correct root per sentence). The dataset consists of over 10,000 labeled training trees and a corresponding test set for evaluation.

II. Data Exploration and preprocessing

The original dataset consists of syntactic trees, each associated with a specific language and a root node label. These trees were flattened into a tabular format where each row corresponds to a node. Given that each sentence has only one root node, the dataset is highly imbalanced.

We computed a variety of graph-theoretic features per node using NetworkX:

- Centrality measures: degree, closeness, betweenness and pagerank. Katz and load centrality were also incorporated to complement the traditional centrality measures. Katz centrality accounts for all paths originating from a node, attenuated by path length, thus capturing indirect influence beyond immediate neighbors. Load centrality, on the other hand, measures how often a node appears in the shortest paths between other nodes, offering a straightforward estimate of how structurally important a word is within the sentence.

- Sentence length (n) and node position.

All metrics were normalized to ensure comparability across trees of different sizes.

Root nodes typically have higher values across all centralities, as it can be seen in the following violin plot. This reinforces the idea that the centrality measures we computed should help the models distinguish root nodes from non-root nodes, as they capture key structural roles that are characteristic of root positions within syntactic trees.

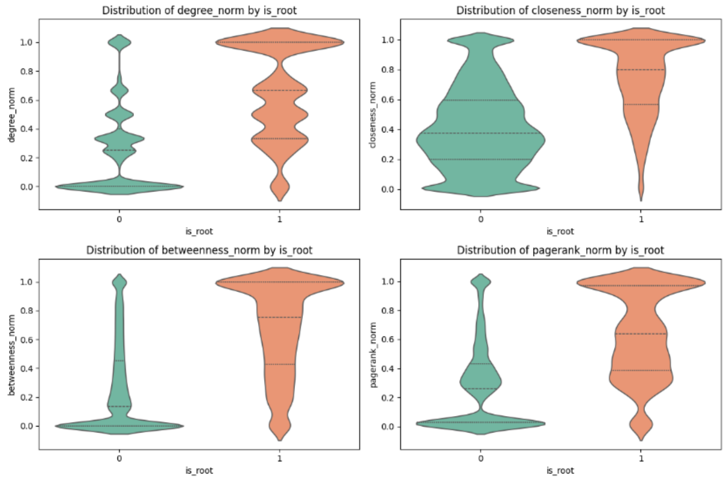

Figure I: Violin plot

We also performed a PCA biplot with the intention of projecting each language based on the average values of the centrality features. The first two principal components capture a substantial proportion of the total variance (72.0% and 24.3%, respectively). Languages that cluster together, like French, Galician, Spanish, and Portuguese, likely share similar syntactic structures in terms of graph topology. In contrast, languages such as Japanese, Korean, and Icelandic appear as outliers, suggesting that the structure of their syntactic trees diverges more strongly from the rest. This is likely to result in considerably lower root prediction accuracy for these languages.

The arrows represent the contribution of each centrality feature to the principal components, where closeness and sentence length are the most influential variables in PC2, while pagerank_norm and betweenness_norm contribute mainly to PC1. This suggests that certain centrality metrics may be more informative for identifying the root in some languages than others. For example, languages like Japanese and Icelandic appear aligned with features such as closeness and sentence length, whereas Romance languages tend to group closer to features like betweenness and pagerank. These differences support the idea that model performance could vary across languages depending on which centrality features dominate their syntactic trees.

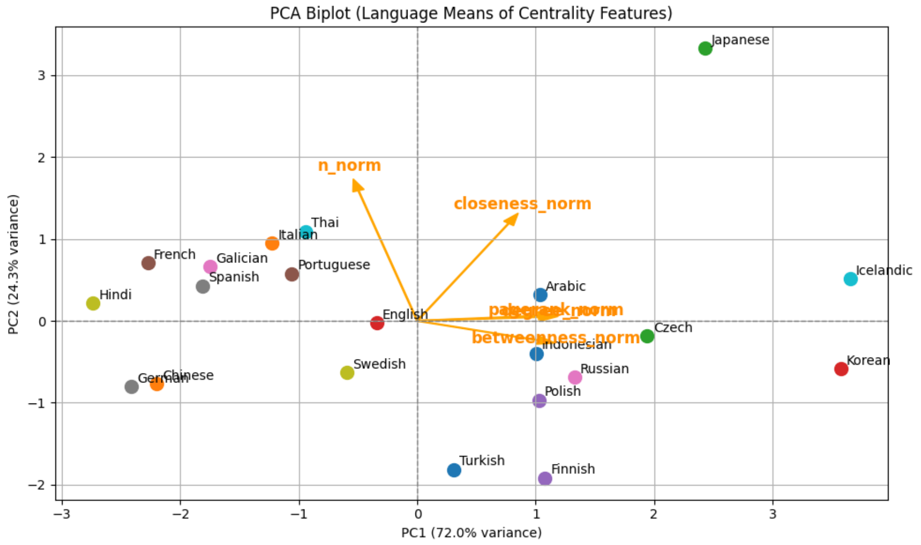

Figure II: Principal Components Analysis biplot

III. Modeling and validation

We framed the task as a binary classification problem: predicting whether each node is the root of its corresponding sentence. For all models, we followed a two-step prediction strategy: First, compute the probability of each node being a root using the trained classifier and second, select the node with the highest probability within each sentence as the predicted root.

To evaluate model performance, we used two complementary metrics:

- Node-based accuracy: the proportion of correctly classified nodes in the entire dataset.

- Sentence-based accuracy: the proportion of sentences where the predicted root matched the true root (considering only the top-probability node counts). This second metric is the most relevant for the task, as only one node per sentence can be correct.

The training data was split using 5-fold cross-validation, where each fold contains disjoint sets of sentence-language pairs. This ensured that all nodes from the same sentence were assigned to the same split, avoiding any data leakage. The splits were generated using 'KFold' with shuffling and a fixed random seed for reproducibility.

The models evaluated include:

- Linear Regression: used only as a simple scoring baseline. It offered a quick benchmark and allowed assessing what scores we were able to obtain with the simplest model.

- Logistic Regression: selected as a strong linear classifier with probabilistic output. It was trained with class_weight='balanced' to address class imbalance and L2 regularization to reduce overfitting. In binary classification tasks, this model should act as baseline.

- Quadratic Discriminant Analysis: included due to its ability to model non-linear decision boundaries. Although sensitive to multicollinearity, it was tested to assess whether generative approaches could exploit the structure of the data.

- Random Forest:

  - One version was trained with default parameters and balanced class weights.

  - Another one used a custom configuration with n_estimators=100, max_depth=10, and class_weight='balanced'.

- XGBoost: We had high expectations for XGBoost, as it was one of the most powerful models studied during the course. The final model used the following configuration: number of estimators = 100, max_depth = 4, learning_rate = 0.1, subsample = 0.8, colsample_bytree = 0.8, reg_alpha = 0.1 and reg_lambda = 1.0. This configuration was selected manually based on performance observed across different experiments and grids.

IV. Results

The final sentence-level accuracies on the held-out test set (with the final version of each model) were as follows:

Model	Accuracy
Linear Regression	0.298
Logistic Regression	0.293
Random Forest	0.257
QDA	0.262
XGBoost	0.314

It is worth noting that the accuracies obtained on the test set are largely consistent with those observed during training. For instance, in the case of Japanese, the best sentence-level accuracy achieved (in the training) was around 5%. One possible explanation lies in the structural nature of Japanese syntax: it is a head-final language, where the root of the syntactic tree often appears at the end of the sentence, in a position that is not structurally central within the graph. As a result, centrality-based features fail to capture the true root position.

Extending this reasoning, it becomes evident that even under optimal conditions, sentence-level accuracies are unlikely to exceed 30% in many cases, due to the limited amount of linguistic information encoded in the available features. Even in more sophisticated natural language processing settings, capturing contextual meaning is a complex task. Therefore, the performance obtained in this project can be considered remarkably good given the minimal and purely structural feature set used.

V. Final model chosen

The final model selected for submission was the tuned XGBoost classifier. It achieved a sentence-level accuracy of 31.4% on the test set, the highest among all models evaluated. This confirms that XGBoost is capable of capturing non-linear relationships between centrality features and the root prediction task.

VI. Scientific and personal conclusions

This project investigated methods for identifying root nodes in undirected syntactic dependency trees, employing engineered graph-based features and a range of machine learning models. The primary goal was to maximize sentence-level root prediction accuracy. Among all models evaluated, XGBoost achieved the highest test accuracy, validating its ability to learn complex patterns from centrality-based features.

From a scientific perspective, the results highlight the value of graph descriptors such as pagerank and closeness in encoding syntactic information. From a practical standpoint, we observed that even simple linear models perform surprisingly well when combined with informative features. This emphasizes the importance of having quality data in any machine learning project, a lesson we will certainly keep in mind for future academic or industry-related work, as good data can entirely shape the expectations and outcomes of a project. This case demonstrates that, in the absence of well-designed features, even simple models can approach the performance of tree-based algorithms like XGBoost.

On a personal level, we believe this project has been extremely useful to consolidate what we learned during the course, and we are already thinking of how to apply these concepts in our professional work beyond the master's programme. Furthermore, the competitive setup, inspired by a Kaggle-like leaderboard, made the project more engaging and offered a benchmark to assess the quality of our results. We would have liked to experiment with model ensembling, letting different models vote on whether a word is the root, but given the clear superiority of XGBoost, it would not have added value in this case. However, we keep this strategy in mind for our next practical machine learning project.