Intelligent Classification of Legislative Proposals

Luca Peres Quinta da Guarda Ronie Paulucio Porfirio

Universidade de Brasília (UnB)

Programa de Pós-Graduação em Computação Aplicada

Course Mineração de Dados Massivos

Prof. Dr. Marcelo Ladeira and Pr. MSc. Gustavo Cordeiro

Contents

Legislative Proposal

- A **Legislative Proposal** is any matter subject to deliberation by the Legislative Chamber (CLDF Internal Regulations, art. 129).

Legislative Proposals can be of the following types:

- Proposta de emenda à Lei Orgânica;
- Projeto de lei complementar;
- Projeto de lei;
- Projeto de decreto legislativo;
- Projeto de resolução;
- Indicação;
- Moção;
- Requerimento;
- Emenda;
- Recursos;

Introduction

Business Understanding - Legislative Proposal's Themes

Legislative Proposals are classified into one or more themes:

- Agricultura
- Assistência Social
- Assunto Fundiário e Ordenamento Territorial
- Assunto Social
- Cidadania
- Ciência e Tecnologia
- Combate à Corrupção
- Comunicação
- Comércio e Serviços
- Cultura
- Defesa do Consumidor
- Desenvolvimento Econômico
- Desporto e Lazer
- Direitos Humanos
- Economia
- Educação
- Energia
- Fiscalização e Governança
- Habitação
- Incentivos Fiscais e Concessões Públicas
- Indústria
- Meio Ambiente
- Não se aplica
- Outro
- Previdência Social
- Relações Exteriores
- Saneamento
- Saúde
- Segurança
- Servidor Público
- Trabalho
- Transporte e Mobilidade Urbana
- Turismo
- Urbanismo

Total: 34 themes

Thematic Classification Benefits

- Efficient classification of legislative proposals is crucial to **streamline their analysis** and processing within the legislative process helping to **determine which committees a proposal should go through**.
- By categorizing legislative proposals into relevant themes, lawmakers can streamline their analysis, **allocate resources efficiently**, and **make informed decisions**.
- This process **enhances transparency** and facilitates a more **organized legislative workflow**.
- Thematic classification plays an important role in maintaining **accurate information retrieval** and **ensuring effective legislative management**.

The Problem

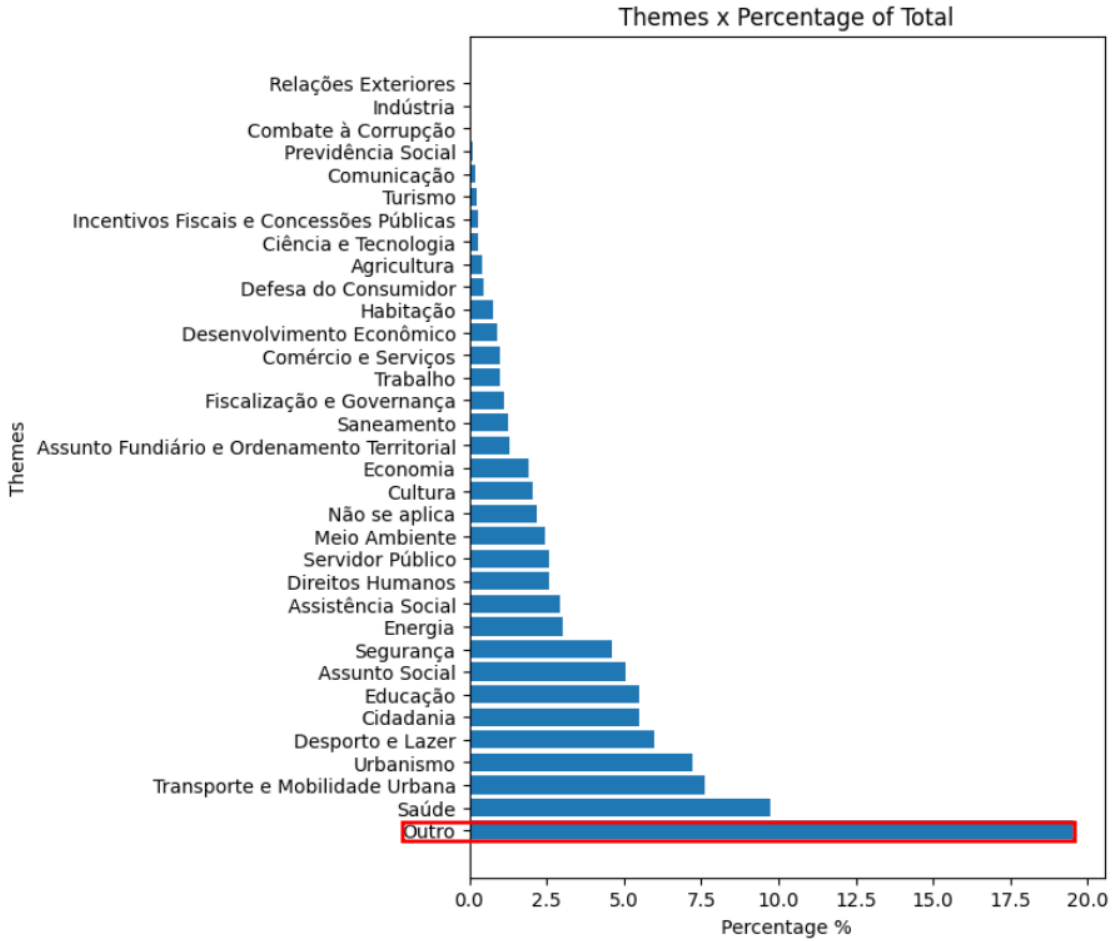
Understanding the problem

Theme “others” is growing bigger

- Usually, the author of the proposal is responsible for classifying it into one or more themes.
- Unfortunately, due to various factors such as ambiguous topics, outdated categories, multidisciplinary nature, **many propositions end up being classified under the generic label of “others”**.

The Problem

Understanding the problem



The chart shows that the number of proposals classified under the theme “others” represents approximately 20% of the total.

The Problem

Problem definition

The problem is inadequate proposal classification

Inadequate classification hinders efficient tracking, analysis, and transparency of legislative activities, making it difficult for both society and lawmakers to understand and oversee the legislative process effectively.

Primary objective

- The primary objective of this study is to **compare different machine learning models** to determine which model is **most effective in suggesting more appropriate theme categories for legislative proposals**.
- The goal is to find **more suitable themes that better match the content of these proposals**.

Objectives

Disclaimer

Attention!

- This study **does not aim to automate the classification process, replace human classification, or compare human classification with machine learning classification**.
- Instead, **the focus is on enhancing the existing categorization process by identifying the best model for suggesting themes for proposals currently classified under the generic label ”others.”**.

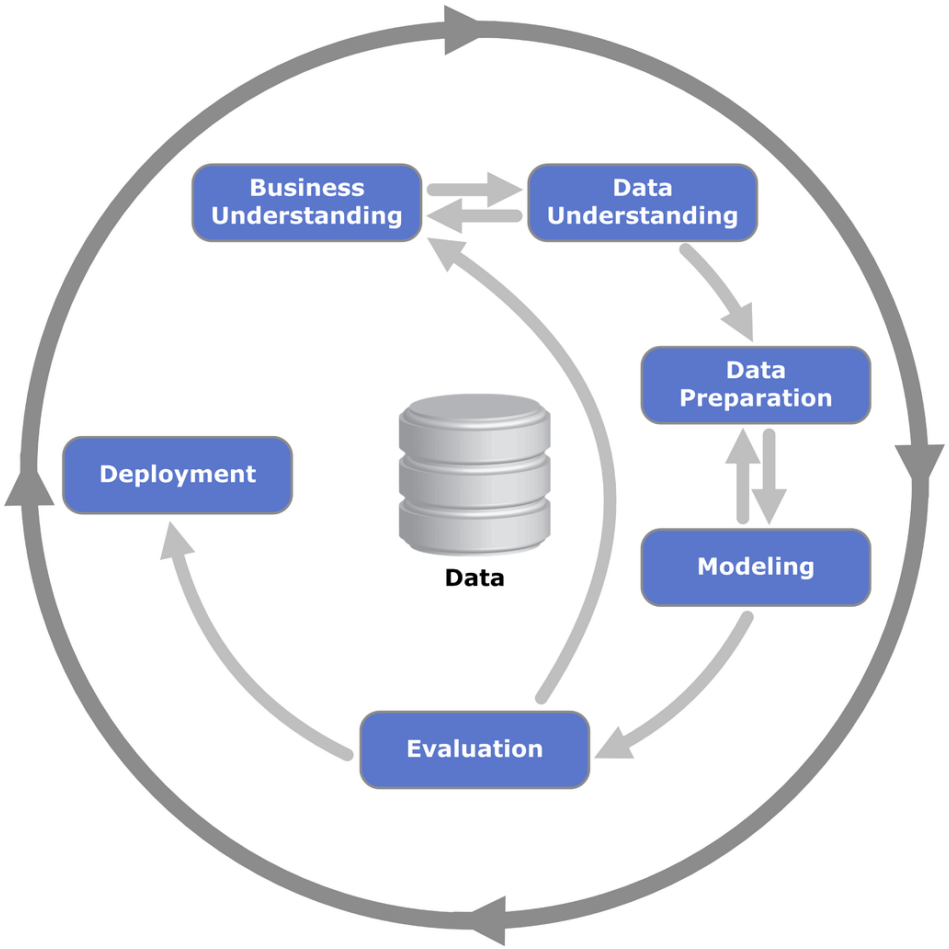
Literature review

Reimers, Nils and Gurevych, Iryna (2019)
Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks
Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing Association for Computational Linguistics

J. Andrade Junior, J. Cardoso-Silva, and L. Bezerra (2021)
Comparing Contextual Embeddings for Semantic Textual Similarity in Portuguese
Anais da X Brazilian Conference on Intelligent Systems

Methodology

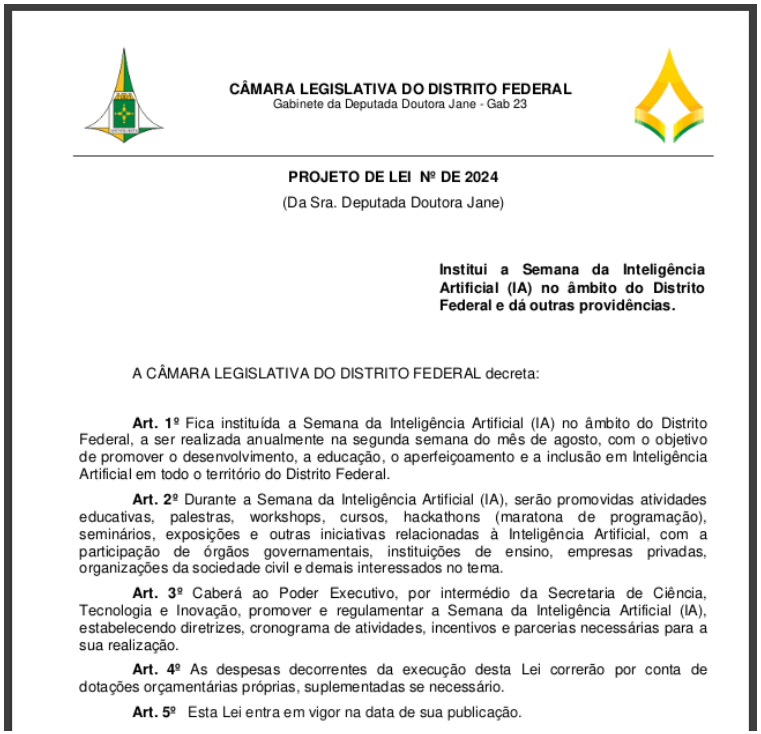
CRISP-DM's Methodology



We utilized the CRISP-DM methodology. Its stages will be explained in the following slides.

1 - Business Understanding

We have analyzed legislative documents to align our data mining objectives with legislative classification needs.



CÂMARA LEGISLATIVA DO DISTRITO FEDERAL
Gabinete da Deputada Doutora Jane - Gab 23

PROJETO DE LEI Nº DE 2024

(Da Sra. Deputada Doutora Jane)

Institui a Semana da Inteligência Artificial (IA) no âmbito do Distrito Federal e dá outras providências.

A CÂMARA LEGISLATIVA DO DISTRITO FEDERAL decreta:

Art. 1º Fica instituída a Semana da Inteligência Artificial (IA) no âmbito do Distrito Federal, a ser realizada anualmente na segunda semana do mês de agosto, com o objetivo de promover o desenvolvimento, a educação, o aperfeiçoamento e a inclusão em Inteligência Artificial em todo o território do Distrito Federal.

Art. 2º Durante a Semana da Inteligência Artificial (IA), serão promovidas atividades educativas, palestras, workshops, cursos, hackathons (maratona de programação), seminários, exposições e outras iniciativas relacionadas à Inteligência Artificial, com a participação de órgãos governamentais, instituições de ensino, empresas privadas, organizações da sociedade civil e demais interessados no tema.

Art. 3º Caberá ao Poder Executivo, por intermédio da Secretaria de Ciência, Tecnologia e Inovação, promover e regulamentar a Semana da Inteligência Artificial (IA), estabelecendo diretrizes, cronograma de atividades, incentivos e parcerias necessárias para a sua realização.

Art. 4º As despesas decorrentes da execução desta Lei correrão por conta de dotações orçamentárias próprias, suplementadas se necessário.

Art. 5º Esta Lei entra em vigor na data de sua publicação.

(a) Proposal example

(b) Proposal's data

2 - Data Understanding

The dataset contains 22,267 summaries extracted from the Electronic Legislative Process (PLE) system, covering the period from 2021 to May 2024. Each summary is accompanied by its respective thematic classification.

Data Preparation

1. **Preprocessing**: We discard data classified under “outro” and “não se aplica” themes. Then, we perform tokenization, normalization, stopword removal, and lemmatization processes.
2. **Vectorization**:
   - **Multilingual sentence embedding model** based on the MiniLM architecture, a lightweight and efficient BERT variant with 12 transformer layers, to produce high-quality embeddings that capture the semantic meaning of the text.
   - **TF-IDF (Term Frequency-Inverse Document Frequency) model** involves converting text into numerical vectors based on the frequency of terms within a document and across a collection of documents. This approach captures the importance of a term in a document relative to the entire dataset.

4 - Modeling

1. **DummyClassifier**: A **baseline model** that makes predictions using simple rules and establishes a baseline to **compare the performance of more complex models**.
2. **Support Vector Machine (SVM)**: A powerful model that **finds the hyperplane that best separates the classes in the feature space**. It is effective for high-dimensional spaces and when the number of dimensions exceeds the number of samples.
3. **Logistic Regression**: A linear model that **estimates the probability of a binary outcome based on input features**. It is simple and interpretable, good for linearly separable data and understanding feature importance.

4 - Modeling (cont...)

4. **XGBoost (Extreme Gradient Boosting)**: An optimized gradient boosting algorithm that **builds an ensemble of weak learners (typically decision trees) to improve model performance**. Known for high performance, speed, and scalability.
5. **Random Forest**: An **ensemble model that constructs multiple decision trees and aggregates their predictions**. Robust against overfitting, good for handling large datasets with higher dimensionality.
6. **K-Nearest Neighbors (KNN)**: A **non-parametric model that classifies a data point based on the majority class of its k nearest neighbors**. Simple and intuitive, effective for small datasets with low noise.

Why Choose These Models?

1. **Baseline Comparison**: DummyClassifier provides a benchmark to gauge the performance of more sophisticated models.
2. **Linear and Non-Linear Data**: Logistic Regression and SVM cover linear relationships, while Random Forest, XGBoost, and KNN handle non-linear data.
3. **Model Performance**: XGBoost and Random Forest are chosen for their strong predictive performance and ability to handle complex datasets.
4. **Interpretability**: Logistic Regression is valued for its simplicity and ease of interpretation.
5. **Versatility**: SVM, Random Forest, and XGBoost offer versatility across different types of data and problems.
6. **Scalability**: XGBoost is particularly chosen for its scalability and efficiency in handling large datasets.

5 - Evaluation

Our evaluation metrics include **Accuracy**, **Precision**, **Recall** and **F1 score**.

- Accuracy is **straightforward and easy to understand**.
- Precision and recall are more informative when dealing with **imbalanced classes** because they provide insights into the performance of the minority class, which accuracy might overlook.
- F1 score gives a **balance between precision and recall as a single metric** to summarize the model performance.

6 - Deployment

- The results might be used to create an interface in the PLE system to suggest themes that best fit new proposals.

Experiments Conducted

nao sei

	Model	fit_time	score_time	test_f1_weighted	test_balanced_accuracy	test_precision_weighted	test_recall_weighted
0	DummyClassifier	0.006086	**0.016813**	0.025819	0.031250	0.014462	0.120258
1	KNeighborsClassifier	**0.005841**	33.463293	0.323486	0.182068	0.447307	0.299509
2	**LogisticRegression**	5.160993	0.024248	0.467718	**0.377732**	0.506144	0.458011
3	RandomForestClassifier	373.229191	1.552665	0.435226	0.273611	0.444914	0.449233
4	SVM	52.730176	14.002061	**0.472803**	0.319393	**0.512542**	0.456292
5	XGBClassifier	128.124318	1.915238	0.449333	0.277803	0.449830	**0.458809**

Table: nao sei

Não sei

- Explicar tabela

Experiments Conducted

Embedding - Cross Validation

Model	fit_time	score_time	test_f1_weighted	train_f1_weighted	test_balanced_accuracy	train_balanced_accuracy	test_precision_weighted	train_precision_weighted	test_recall_weighted	train_recall_weighted
DummyClassifier	0.028611	0.013967	0.025819	0.025819	0.031250	0.031250	0.014462	0.014462	0.120258	0.120258
KNeighborsClassifier	0.040055	1.049414	0.307679	0.516615	0.179927	0.370583	0.323286	0.547667	0.311971	0.520135
LogisticRegression	3.597823	0.025262	0.252122	0.282075	0.217896	0.387424	0.356856	0.397779	0.234561	0.270780
RandomForestClassifier	225.171689	0.825105	0.299710	**0.807638**	0.163765	**0.865793**	0.328777	**0.835075**	0.327624	0.801934
SVM	42.684947	33.030448	0.295933	0.346580	**0.241816**	0.495388	**0.376351**	0.435997	0.276796	0.337201
XGBClassifier	253.781450	0.560534	**0.325231**	0.800150	0.178569	0.726839	0.330270	0.800984	**0.341866**	**0.802977**

Table: Embedding - Cross Validation

Embedding

- Explicar tabela

Experiments Conducted

TfidfVectorizer - Cross Validation

Model	fit_time	score_time	test_f1_weighted	train_f1_weighted	test_balanced_accuracy	train_balanced_accuracy	test_precision_weighted	train_precision_weighted	test_recall_weighted	train_recall_weighted
DummyClassifier	0.005016	0.013737	0.025819	0.025819	0.031250	0.031250	0.014462	0.014462	0.120258	0.120258
KNeighborsClassifier	0.004096	28.906825	0.323249	0.541327	0.181929	0.396655	0.446675	0.629905	0.299141	0.527901
LogisticRegression	3.725738	0.018837	0.467718	0.584556	**0.377732**	0.727172	0.506144	0.629863	0.458011	0.585697
RandomForestClassifier	267.923079	1.148535	0.435199	**0.806360**	0.272152	**0.864809**	0.445313	**0.834005**	0.449355	**0.801044**
SVM	44.651407	12.235956	**0.472803**	0.676972	0.319393	0.803307	**0.512542**	0.712173	0.456292	0.673389
XGBClassifier	102.740959	1.606697	0.449333	0.766095	0.277803	0.768935	0.449830	0.767603	**0.458809**	0.769460

Table: TfidfVectorizer - Cross Validation

TfidfVectorizer

- Explicar tabela

Experiments Conducted

TfidfVectorizer and MultiOutputClassifier - Cross Validation

Model	fit_time	score_time	test_f1_weighted	train_f1_weighted	test_balanced_accuracy	train_balanced_accuracy	test_precision_weighted	train_precision_weighted	test_recall_weighted	train_recall_weighted
DummyClassifier	0.004188	0.011938	0.025819	0.025819	0.031250	0.031250	0.014462	0.014462	0.120258	0.120258
KNeighborsClassifier	0.005413	29.213731	0.323249	0.541327	0.181929	0.396655	0.446675	0.629905	0.299141	0.527901
LogisticRegression	3.685860	0.019390	0.467718	0.584556	**0.377732**	0.727172	0.506144	0.629863	0.458011	0.585697
RandomForestClassifier	268.849362	1.154890	0.434992	**0.806396**	0.272071	**0.864844**	0.444953	**0.834183**	0.449048	**0.800982**
SVM	44.404848	12.142895	**0.472803**	0.676972	0.319393	0.803307	**0.512542**	0.712173	0.456292	0.673389
XGBClassifier	102.840638	1.657798	0.449333	0.766095	0.277803	0.768935	0.449830	0.767603	**0.458809**	0.769460

Table: TfidfVectorizer and MultiOutputClassifier - Cross Validation

TfidfVectorizer and MultiOutputClassifier

- Explicar tabela

Conclusions

- Quais?

Future Work

Future Work

- Quais?