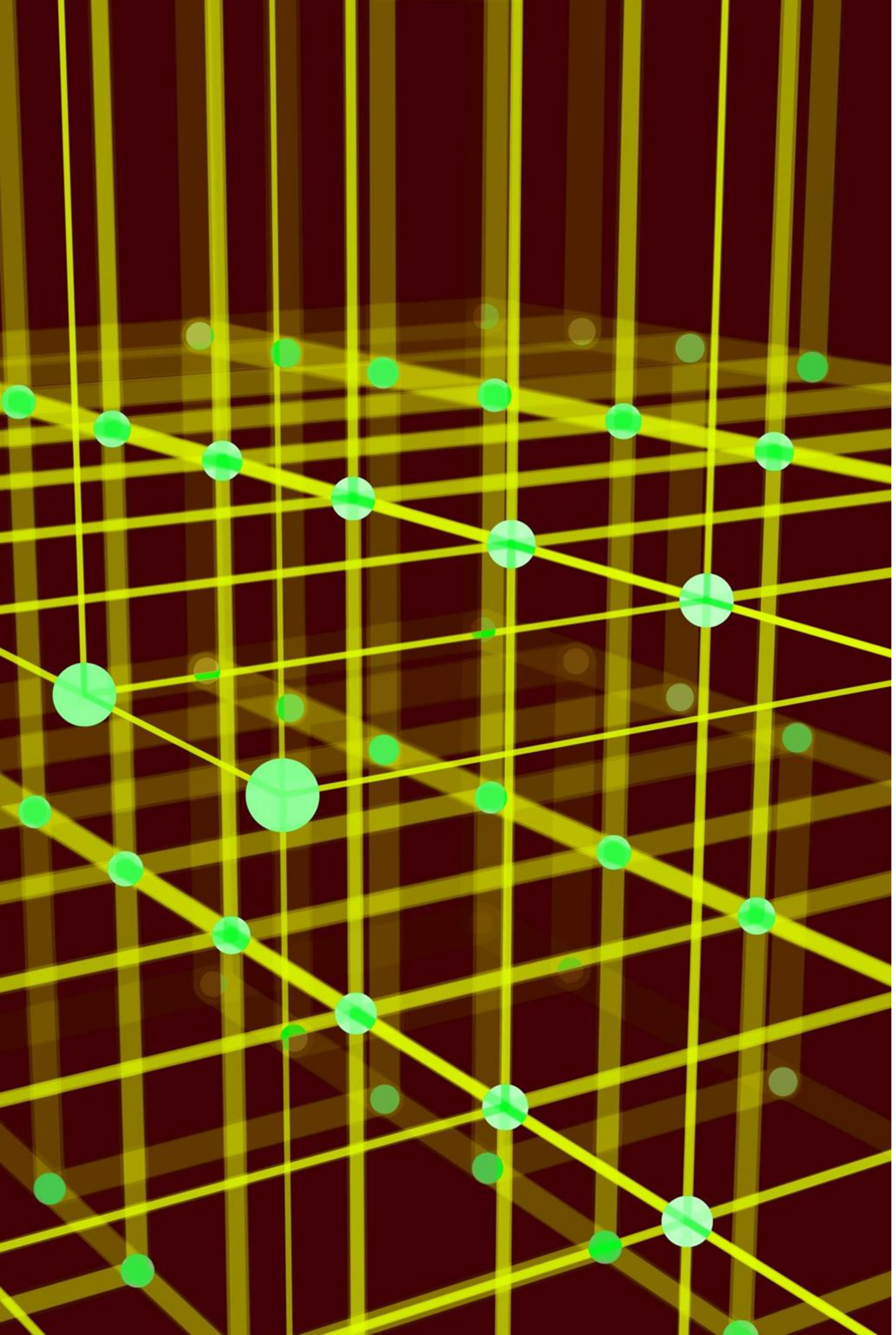


Domain Name Classification Challenge

Data Challengers

Professor: Giannis Nikolentzos

Contents

- Dataset Overview
- Graph
  - Feature extraction
  - Classification
- Text
  - Pre - processing
  - Feature extraction & selection
  - Class imbalance & missing text
  - Classification
  - Classification results
- Text & Graph combination

Dataset Overview

We were given a part of the Greek web **Graph**.

- 65k total Nodes (domains).
- 1.6M Directed Edges.
- 9 Classes.
- 41k nodes come with text attached.
- Training Set of 1800 labeled samples.
- Test Set of 605 samples to predict.

Graph

Feature extraction

We tested multiple methods to extract node features. Some of them are:

- Graph Attributes (out degree, in degree etc.)
- Random Walks
- Node2Vec
- SDNE

The best performing features were Random walks with 30 walk length and 200 random walks.

Classification (1/3)

Graph Convolutional Network (GCN)

- ❖ 2convolutional layers
- ❖Dropout layers
- ❖Batch Normalization layer
- ❖Classification layer with Softmax
- ❖Adam optimizer
- ❖Skip connections.

Classification (2/3)

GraphSAGE

- 2 convolutional layers
- Dropout layers
- Batch Normalization layers
- Classification layer with softmax
- Adam optimizer
- Skip connections

Classification (3/3)

- **Benchmarking results on Graph classification.**

Features	Classifier	Private score	Public score
Random Walk	GCN	0.85	**0.75**
Random Walk	GraphsAGE	**0.81**	0.78

Text

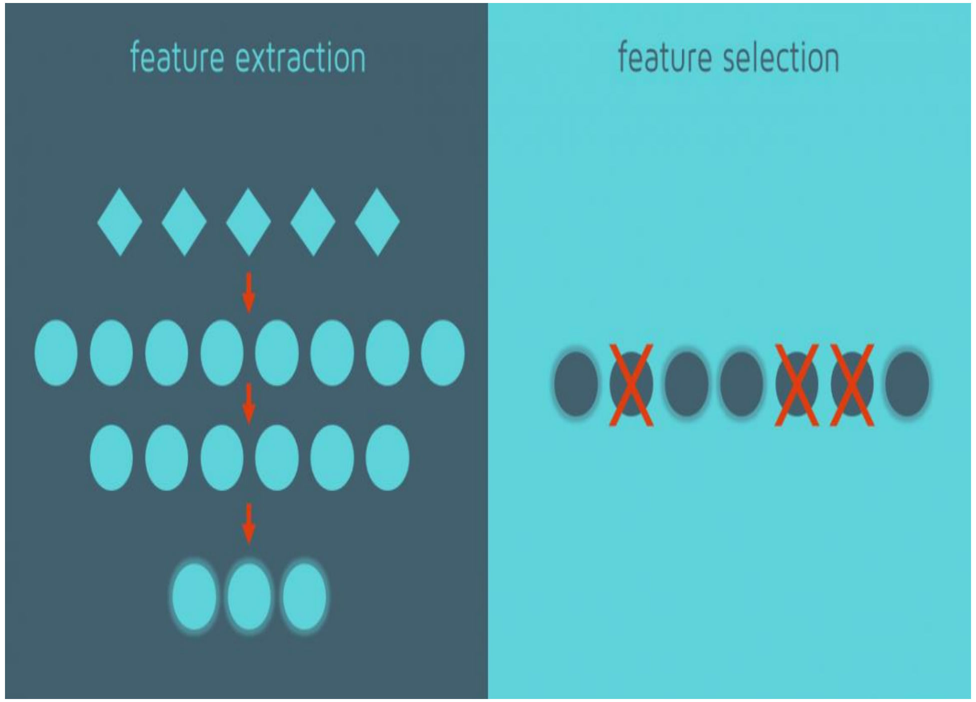
Pre - processing

- Converted to lowercase.
- Removed accents, tones.
- Removed punctuation & numbers.
- Stop words removal.
- Lemmatization of tokens.
- Hyper- links removal.

Feature Extraction & Selection

We used:

- ❖TF-IDF
- ❖FastText
- ❖ Doc2Vec
- ❖ Bert Tokenizer



Class Imbalance & Missing Text

Upon Analyzing the training set:

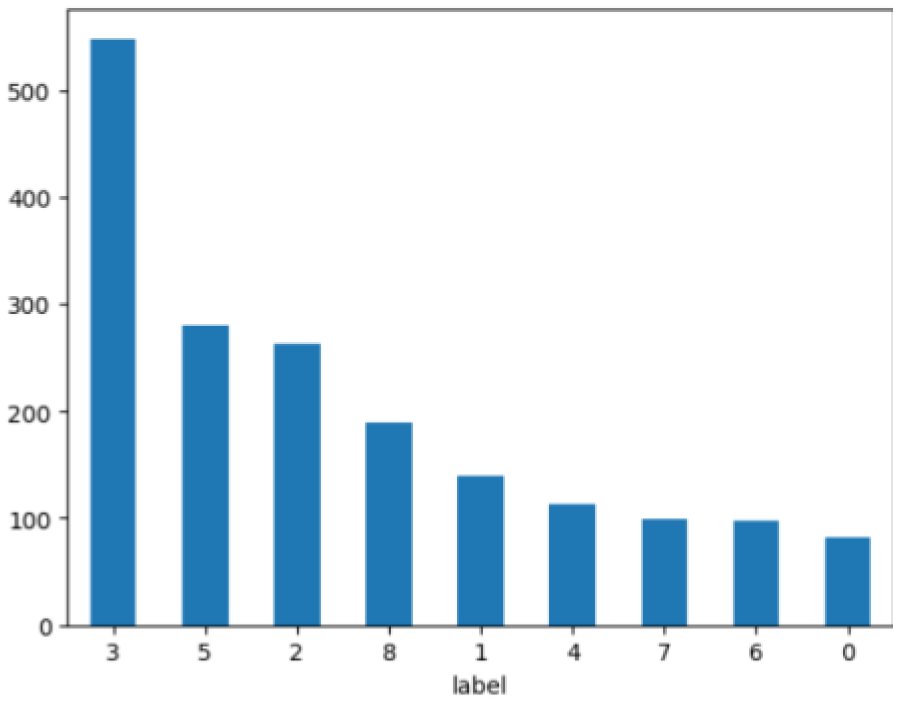
- class imbalance was evident

Upon Analyzing the test set:

- 98 text instances were missing

Disadvantages :

- Difficulty to train nonbiased classifiers
- Missing text -> hindering the performance of CLFs



Classification (1/3)

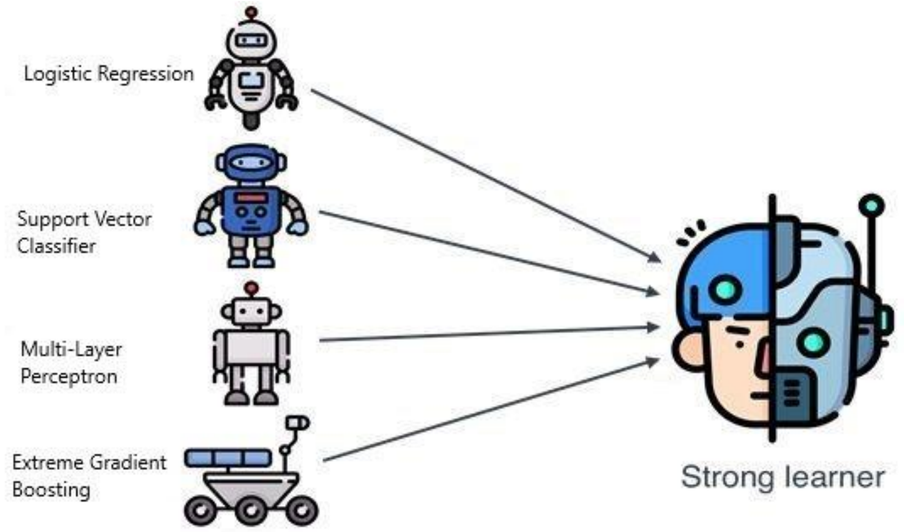
For the classification part, a variety of ML algorithms were tested:

- TF-IDF
  - Lemmatization improved the scores significantly.
  - Max features -> 5000
  - SVD
  - Logistic Regression
  - Naïve Bayes
  - Average predictions between classifiers.

Classification (2/3)

For the classification part, a variety of ML algorithms were tested:

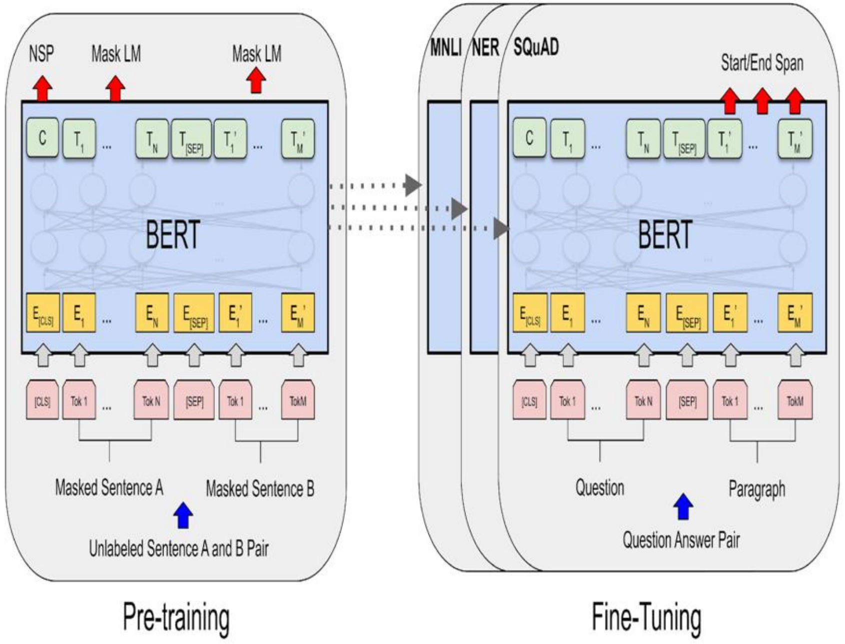
- FastText (pre-trained)
  - Cross-Validated Logistic Regression
  - CV Support Vector Classifier
  - CV Extreme Gradient Boosting
  - MLP-Classifier with fully connected layers
  - All the above as ensembled technique



Classification (3/3)

For the classification part, a variety of ML algorithms were tested:

- BERT
  - Due to the large text size of each domain we splitted the texts to subtexts.
  - We tried to use pseudo labeling technique, using the predictions of GCN with over 0.99 probability.
  - Better results with 300 token window size.



Text Classification Results

Summarizing results about the classifiers we tested:

Features	**Classifier**	**Private score**	**Public score**
TF-IDF	Logistic Regression & Naïve Bayes	1.11	1.15
FastText	MLP	**1.07**	1.16
BERT pre-trained embeddings	BERT	**1.07**	**1.05**

Text & Graph combination

We conducted several trials with:

- **Stacking**
  - Which was unstable and more complicated.

- **Average** predictions between the 2 models.
- We used GCN only predictions for the domains with missing text.

Text & Graph combination Results

- Our Final results

Features	Classifier	Submitted	Private	Public score
GCN	BERT with augmented data	No	0.73	0.69
GCN	BERT	**Yes**	0.74	0.70
GCN	TFIDF	**Yes**	0.75	0.70