Citation Prediction Challenge

ALTEGRAD 2021/2022

Marine Mercier, Yoach Lacombe

MVA 21/22
ENS Paris-Saclay

February 22, 2022

Overview

Overview of the pipeline

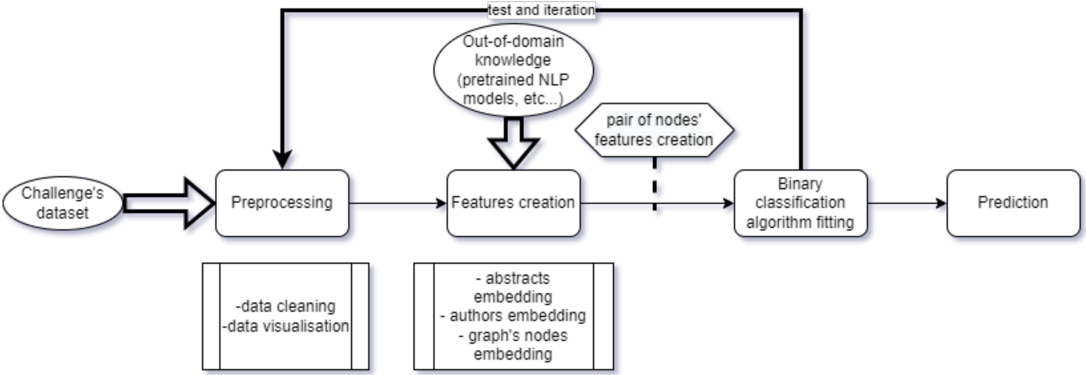

Figure: Challenge's pipeline.

Simple features - 1

Baseline

1. Sum of degree
2. Absolute difference of degree
3. Sum of the length of the abstracts
4. Absolue difference of the len of the abstracts
5. Number of common words in the abstracts

Authors related

1. Number of common authors
2. Number of common authors weighted by number of listed papers in which the authors appeared
3. Number of authors
4. Popularity of each paper (sum of appearances of each author)

Simple features - 2 - graph related

Graph related features

$\Gamma(u)$ denotes the set of neighbors of u.

- Jaccard Coefficient - $\frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$
- Ressource Allocation index - $\sum\limits_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{|\Gamma(w)|}$
- Adamic Adar Index - $\sum\limits_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{log|\Gamma(w)|}$

Testing difficulties

These similarities measures heavily rely on the graph structure. Splitting the positive edges when testing modify this structure, which makes the testing difficult.

Limitations and overcoming

These features are **hand-made**, based on simple intuitions :

- Papers are more likely to be linked if they have similar contents (based on abstracts).
- Famous or similar authors are more likely to quote each other.

To get better results, our features need to be robust to **neighborhood context** and **semantic meaning**.
We thus looked at **NLP** and **graph-based** algorithms.

Abstract embedding - Doc2Vec

Doc2Vec

1. Extension of the classic Word2Vec algorithm (CBOW approach)
2. Previous words AND document related embeddings are used to predict the current word (PV-DM).
3. Trained with Gensim directly on the abstracts

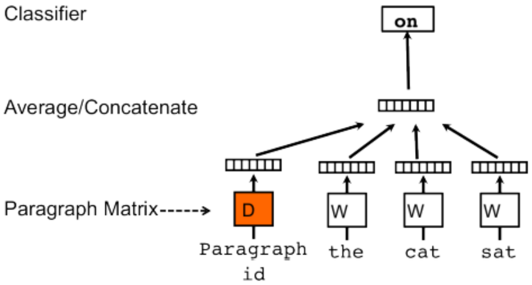

Figure: Visual illustration of the doc2vec's PV-DM approach. From https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e

Abstract embedding - USE

Universal Sentence Encoder

1. Transformer-based encoder.
2. Pre-trained on multiple tasks (skip-thought, response prediction, natural language inference) on a large dataset.
3. Wasn't fine-tuned on the abstracts - use of out-of-domain knowledge.

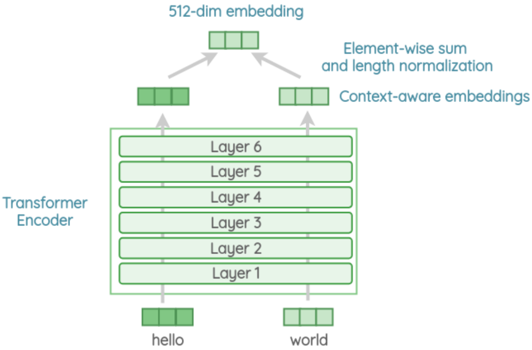

Figure: Transformer mechanism. From https://amitness.com/2020/06/universal-sentence-encoder/

Authors embedding - Node2vec

Node2Vec

1. Sample sentences of nodes with random walks starting from each node.
2. Use word2vec algorithm on sentences of nodes to create node embeddings.

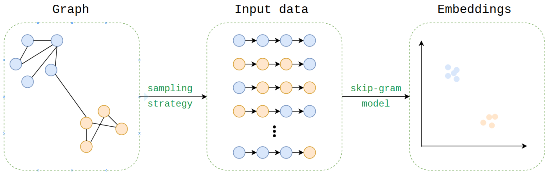

Figure: Node2vec framework. From :"node2vec: Scalable Feature Learning for Networks", Aditya Grover and Jure Leskovec

Graph-based embeddings - GraphSAGE

GraphSAGE framework

On a graph with node features. K aggregators functions, W^k weight matrices.

1. Sample uniformly a fixed-size set among neighbors of nodes in the considered layer.
2. Aggregate features of each layer's nodes with the aggregators.
3. Use a fully connected layer on the concatenation of features of each layer with weights W^k.

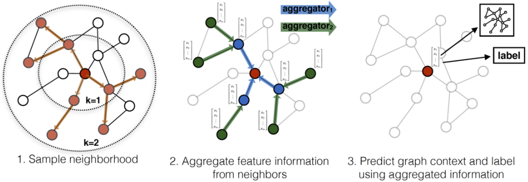

Figure: GraphSAGE framework. From : Inductive Representation Learning on Large Graphs, William L. Hamilton and Rex Ying and Jure Leskovec

Building the binary dataset

Negative edges sampling

To create a binary dataset, we need negative samples. We sampled pairs of nodes as long as they weren't on the positive edges set.

Negative to positive edges ratio

We used a negative to positive edges ratio of around 1.

From unstructured dataset to tabular dataset

Two categories of features:

- Node-based features - either using similarity functions (l2, l1, cosine similarity) or concatenating them.
- Edge-based features - used as it is.

Binary classifier

Input : Edge embeddings
Output : Link prediction

Neural Network

- Three linear layers with ReLu activation function,
- Dropout to reduce over-fitting,
- Sigmoid function for the output,
- Trained simultaneously with GraphSAGE.

XGBoost

- Random forest technique : bagging of shallow trees trained on a sample of the dataset, with a random selection of features.
- At each iteration use the error residuals of the previous model to fit the next model.
- Average all trees to predict the output : $\widehat{y} = \sum_{i=0}^{m} \alpha_i F_i(X)$ with F_i the learned trees, α_i the learning rate at iteration i.

Results on train

Results after training on 70% of the dataset, 30% for validation. With XGBoost classifier.

Features used	Accuracy	Log Loss	AUC
Authors based features	0.776	0.501	0.629
+ baseline features	0.898	0.253	0.940
+ USE embedding similarity	0.919	0.208	0.960
+ Doc2Vec embedding similarity	0.931	0.178	0.971
+ SAGE embedding similarity	0.961	0.110	0.989

Table: Results on train/test split with 70 percent of the edges on the train set.

Our different features improved all metrics we used to evaluate our pipeline.

Conclusion and next steps

Conclusion

- All in all, our best model, trained on the full dataset,led to a **public score of 0.12626**.
- The **GraphSAGE** embedding (which encompass node features) and the **popularity of the authors** were the most important features.
- The structure of the graph, and the number of connected components impacts some similarity measures more than others.

To go further

- Our model is inductive and would generalize well on new nodes in an evolving graph.
- We could use the entire provided dataset to train our model using the similarity measures impacted by the train/validation split.

Annexe - Feature importance

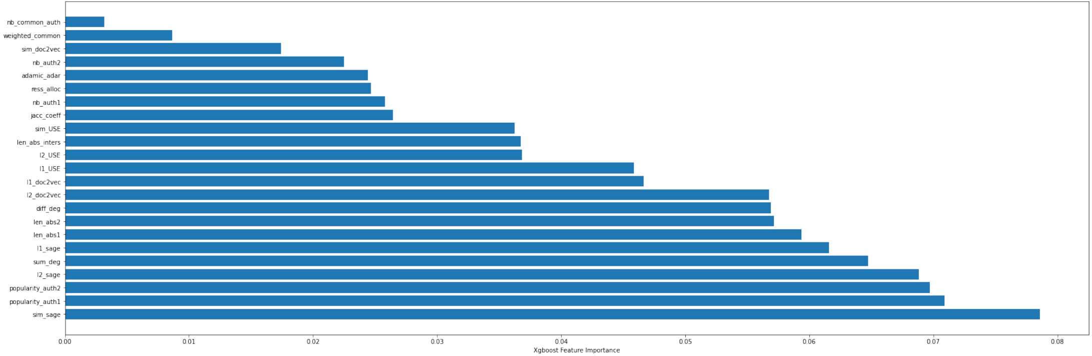

Figure: Features importance by "weight".

The End