DATA CHALLENGE: PREDICTION OF MISSING LINKS IN A CITATION NETWORK

Léo Bouloc [1, 2] *Cyril Gaudefroy* [1, 2] *Ariel Shemtov* [1, 2] *Thai-Chau Truong* [1, 3]

[1] ENS Cachan, 61 Avenue du Président Wilson, 94230 Cachan
[2] ENSTA ParisTech, 828 Boulevard des Maréchaux, 91120 Palaiseau
[3] Télécom ParisTech, 46 Rue Barrault, 75013 Paris

ABSTRACT

This report presents our work on the data challenge for the "Advanced Learning on Text and Graph Data" (ALTeGraD) the purpose of which was to predict missing links in a citation network. From the dataset, we construct a set of features on two groups: text and graph. Then, we perform the classification task by random forest to find out features that contribute the most positively to the solution of the problem.

1. INTRODUCTION

The citation network that we consider here is made of research papers at its nodes, and links between nodes where one of them cites the other. The common information available for each paper in the network is the following: title, authors (some with affiliation), year published, publisher and the abstract.

2. FEATURE ENGINEERING

For our learning strategy, we compute features to describe the relationship between each pair of nodes in the citation network. The features belong to two levels text and graph.

2.1. Text level: Shared relevant words

The reason to use this feature is based on a fact that two papers citing each other often belong to the same research field and therefore share the same set of some keywords. This feature is constructed mainly on the abstract field.

2.1.1. Text preprocessing

- **Removing NaN fields**: The original dataset contains some NaN values. We replace these fields by empty fields. These values will generally not be used for the feature construction and classification processes.
- **Removing stopwords**: To distinguish scientific keywords with regular words in English, we first did a preprocessing step that eliminates all stopwords using an available corpus of sklearn.

2.1.2. Text features

We construct two types of features concerning common relevant words:

- **Bag-of-words model**: a matrix with lines corresponding to documents and columns to words (ignoring stopwords) is created. The values in this matrix stand for how many times each word is present each document. In order to introduce a notion of word weight as first proposed in [2], each column is then normalized so that it sums to one. Now, to compute the similarity between two documents, we used the intersection kernel, which computes the sum of the element-wise minimum between two line vectors.
- **Term Frequency-Inverse Document Frequency (TF-IDF)**:

2.2. Graph level

2.2.1. Number of shared citations

In a citation network, this quantity represents the number of common neighbors in the network. In other words, each neighbor is a distinct paper that is linked to both nodes of the considered pair. Our assumption is that when two papers have a number of common citations, their subjects are more closed to each other and the probability for them to be cited to each other will be higher. As described in section 3, this appears to be the most important feature.

2.2.2. Difference in publication time in the same journal

For this feature, we include two characteristics which are: whether two papers belong to the same journal and their difference in the publication time. The assumption behind this is that in the same journal, the subjects of two papers are somehow related to each other because they are in the same field. And the probability that two papers have a link depends also on the publication time. In section 3, it shows that this feature doesn't have much impact comes from the fact that the network that we consider is non-oriented.

2.2.3. Number of common authors

The number of shared authors between two articles is a very significant indicator, although shared authors don't happen in many pairs.

3. MODEL TUNING AND COMPARISON

3.1. Experiment method

The strategy that we chose is a random forest. The advantages of this method is that it provides feedback on which features perform best, and also it tends to avoid over-fitting. The model is trained on the training set, and then used on the test set to predict whether there exists a link or not. Since it takes too long to perform a cross-validation test, we only divide the original dataset into two parts:

Table 1. Performance on each separate feature groups (in %)

Feature	Training error	Testing error
Common keywords using BoW	99.67	86.05
Common keywords using TF-IDF	83.16	82.94
Number of common citations	98.81	**93.54**
Publication date difference	64.47	64.28
Number of common authors	83.16	82.94

75% of the data is used for training and 25% is used for testing. Although this type of testing error is not a very good estimator for the generalization error on the final result, it could still help us reduce the time to test the quality of each feature set. We use both training and testing error also to examine if there is over-fitting or under-fitting situation on the data.

3.1.1. Result on each individual type of features

The goal of this experiment is to evaluate the significance of each separate type of features. Table 3.1.1

3.2. Tuning parameters on random forest

4. OTHER EXAMINED METHODS

Another problem that needs to be addressed is how to cluster synonyms. On all abstracts, there are words that are synonyms of each others. They may have similar meaning (e.g. problematic, troublesome) or in sigular/plural forms (e.g. package, packages.) It could be imperative to determine cluster the words that have similar meaning. However, one major problem is that the dataset does not contain a sufficiently large amount of text to feed into a Word2Vec mode. Therefore, an alternative solution that we tried was to reuse a Word2Vec model that was pre-trained by Google in [1]. But this type of feature still did not work well on the classification task. (The training and testing accuracy for this feature was under 70%)

5. CONCLUSION

6. REFERENCES

[1] "Word2vec: Google code project," 2013.

[2] Karen Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.