DATA CHALLENGE: PREDICTION OF MISSING LINKS IN A CITATION NETWORK TEAM: TRUONG-BOULOC-GAUDEFROY-MIKHOV

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 course "Advanced Learning on Text and Graph Data" (ALTeGraD). The purpose 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 some machine learning classifiers 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. We began with the construction of some useful features based on these information and then chose the feature combinations that yield the highest performance. To classify, we utilize some well-known learning model such as random forest (RF) and support vector machine (SVM) with linear and radial basis function (RBF) kernels.
The outline of this report is as following: In Section 2, we present the features that we chose to construct for texts and graphs. The experiment methods are available in Section 3. Section 4 states some other methods that were tried but did not work well and also possible extensions. Finally, we give our conclusion in Section 5.

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. 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. 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 [1], 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)**: We also constructed two TF-IDF features ([2], [3]) with the n-grams of 1 and 2. In the experiments, we compare the performance of two types of feature at the text level.

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 reason to use these features is that an author is usually specialized in one or more fields. Therefore, his research is continued in many papers. Specifically, this is a group that contains three main features:

- The number of common authors in two papers
- The number of authors in paper 1 that appear in the abstract of paper 2 and vice versa.

The first feature is more important because an abstract usually does not mention the name of the authors. In the experiment part, it shows that the number of shared authors between two articles is not itself

a strong indicator for link prediction. On the contrary, when combining with other features at two levels, this feature can enhance the accuracy.

3. MODEL TUNING AND COMPARISON

3.1. Experiment method

To classify the features, we chose random forest and SVM with linear and RBF kernels (The polynomial kernel was in our choices from the beginning but it was outperformed even by random forest. Therefore we do not present the result with this kernel.) 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.
Due to the resource limit, it takes too long to perform a cross-validation test. We use the classical error criterion by dividing the original dataset into two parts: 75% of the data is used for training and 25% is used for testing (in the code, this is called "development mode" with a flag variable). Although this type of testing error is not a very good estimator for the generalization error on the final result, it could 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.

3.2. Method to combine features (feature selection)

We use the following ad-hoc approach to determine the best set of features:

- First, we estimate the contribution of each separate group of features mentioned in Section 2.
- Second, for less significant features, we group them together to see if we could have a more robust one and combine with important features.
- From the important features, we combined some of them together and see if the accuracy of the model could be ameliorated.

This process was carried out manually to that we could control the sets of chosen features. The experiment section points out clearer the best combinations that we managed to capture.

3.3. Tuning parameters for classifiers

Beside the choice of features, the parameters of classifiers are also quite important. Our method of tuning parameters for the two chosen classifiers is as following:

- For the random forest, the more available estimators will lead to a model with lower variance. Therefore, we do experiments on the number of estimators and fix other settings. In Section 3.4, it shows that when the number of estimators increases, the accuracy converges to a certain value. The best random forest model is chosen as the one with the least number of estimators that achieves this converged accuracy.
- For the SVM, we chose heuristically the regularization parameter C=1.0 on all experiments. There is only one parameter set for linear kernel and it is compared with RBF kernel. The parameter γ in the RBF function $exp(-\gamma||x-y||^2)$ because with different feature sets, the L2-norm in this function varies significantly. When the value of γ is too small, the model becomes too sensitive to the data, which is because changing one instance will lead to the effect on all data points. Contrarily, when γ is too large, the model tends to take all points as support vectors because the change at one point will not have any effect on any other point. Therefore, it is imperative to have an appropriate γ for each set. In Section 3.4, we list the best classifier for each mentioned set of feature.

3.4. Experimental results

3.4.1. Result on each individual type of features

Table 3.4.1 shows the result for each separate feature group.
We can see that the number of common citations is the most valuable criterion even when not being combined with others.

3.4.2. Model tuning

3.4.3. Comparison of classifiers

3.4.4. Feature selection

4. SOME METHODS THAT DID NOT WORK OUT

Here is some methods that need to be extended or that we tried but did not work well. We list here to remark and see if others managed to apply or if they could be improved in the future.

- **Addressing the synonyms by Word2Vec:** A problem that needs to be addressed is how to cluster synonyms. In 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 [4]. But this type of feature still did not work well on the classification task. (The training and testing accuracy for this feature was under 60%.)
- **The use of Doc2Vec [5]:** We also try to capture some correlation between two document in each pair by a Doc2Vec model whose library is available in gensim 0.10.3. This approach also did not work out. A possible explanation is that at the level of text meaning, two papers that are cited in a pair do not share much correlation. Our use of Doc2Vec could not capture much difference between a cited and non-cited pair.
- **Citation link prediction of old and theoretical papers:** As mentioned above, our prediction of a link is mainly based on the similarity in a text and in a network. In reality, there may be the case when we have a paper which is quite old and is in a completely different domain but contains a significant theory for the current paper that cites to. This is the case that we could not handle well because almost all features that we chose favor the papers that are closely related to each other.

5. CONCLUSION

In the scope of this project, we managed to build a model that predicts the links based on the choice of appropriate features and the use of random forest. The parameters are chosen based on the common information of a paper at the level of graph and the common

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

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

keywords on the text level. The criterion to choose the feature is the contribution to the performance of the system. Final result shows that graph features play a significant role in link prediction and must be present in the final chosen features.

One of the drawback of our approach lies at the feature selection. Due to the need for determining the best classifier for each feature set, this process is almost infeasible to conduct completely automatically. We hope to find a better solution in the near future.

6. REFERENCES

[1] 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.

[2] Amit Singhal, Chris Buckley, and Mandar Mitra, "Pivoted document length normalization," in *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 1996, SIGIR '96, pp. 21–29, ACM.

[3] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok, "Interpreting tf-idf term weights as making relevance decisions," *ACM Trans. Inf. Syst.*, vol. 26, no. 3, pp. 13:1–13:37, June 2008.

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

[5] Quoc V. Le and Tomas Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014.