CS-433 Machine Learning - Project 2

Yan FU[a], Shengzhao XIA[a], Runzhe LIU[b]
[a] *Institute of Electrical Engineering, EPF Lausanne, Switzerland*
[b] *Institute of Mechanical Engineering, EPF Lausanne, Switzerland*

I. Introduction

For this project we joined Chair of Economics and Management of Innovation. Since July 2016, thousands of researchers have been dismissed after the Erdogan government repressed the coup against it. Provided with the complete dismissal list of Turkish researchers we tried to find out what could be possible factors affecting one's dismissal.

The pipeline of our project can be simply listed as below:

- Data collection using web scraping from web of science and basic data cleaning. We collected over 350 thousand paper records from web of science (WOS) and saved them in the format of .csv. Information includes the title, author information, language, publish year, journal, key words, subject and uid.
- Name disambiguation. To collect complete paper record of a certain author, we search the abbreviated name rather than full name of authors. And this will definitely cause name disambiguation problem (different researchers using same name).
- Setting up and testing several machine learning models, comparing the performance and explaining the effects of different features.

II. Name Disambiguation

The data is obtained by web scraping using extended version of web of science. The strategy is to search the full family name + first letter of given name (eg. For a man named Tom Smith we search Smith T*). And we searched for over 3000 Turkey researchers in the dismissal list. With this strategy, WOS will return far more records than just searching Smith Tom. But it will arise another important problem: the name disambiguation. Name disambiguation can occur when one is seeking a list of publications of an author who has used different name variations and when there are multiple other authors with the same name. So before we proceed to classification, we need to first distinguish the correct author of each paper (e.g. Paper written by Smith Tom and Smith Tim).

To do disambiguation on papers of authors with the same abbreviated name, we are going to use more information other than author name and affiliation. And usually a certain author has his specific research field, the subject of each paper can be used as a good feature to realize the disambiguation.

A. word2vec

The word2vec takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary from the training text data and then learns vector representation of words. And we can use the word vectors as features in machine learning.

Full name	Affiliation	Papers Found
DOGAN METIN	INONU UNIV	26
DOGAN MEVLUT	AFYON KOCATEPE UNIV	42
DOGAN MUAMMER	AKSARAY UNIV	0
DOGAN MUSTAFA	PAMUKKALE UNIV	10
DOGAN MUZAFFER	ANADOLU UNIV	1

TABLE I: Dismissed researchers' record found in WOS

Google offers pre-trained vectors trained on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases. However, the google archive is less academic and many keywords are not in vocabulary. So we decided to create our own keyword corpus and train the keyword model. A simple way to investigate the learned representations is to find the closest words for a specified word. To test the performance of our model, we show an example of a given word parkinson disease:

```
('medial prefrontal cortex', 0.9935349225997925),
('brain imaging', 0.9935184121131897),
('motor coordination', 0.9932699799537659),
```

Fig. 1: Similar words of 'parkinson disease'

And the close words are highly related to parkinson disease(eg.brain imaging is widely used in PD diagnosis) so the model is good.

B. Pre-processing

In our data set, we have over 350,000 papers and in the dismissal list we have more than 3000 authors in full name. And we test the clustering algorithm on one abbreviated name 'DOGAN M' as an example.

In dismissal list there are 6 researchers under same abbreviation name 'DOGAN M'. Table 1 shows their full name and affiliations and records found. And we have 150 papers unlabelled (papers with no affiliation found). The records are found in the whole data set by searching papers with matches of author name and affiliation (eg. The 21 papers in first column include papers by Dogan Metin from Inonu Univ. and Dogan M from Inonu Univ.) And it's possible that several researcher share the same name and affiliation. And we check this by point wise euclidean distances of each papers' subject to the mean central. The idea is simple: we want to find the most relevant subjects of an author and ignore papers in far

subjects. If the distance of subjects of some papers are far away from average, we will consider these papers are written by different researchers. The histogram plots of euclidean distance of subjects tell us that the basic disambiguation (only matching fullname and affiliation) works well on 'DOGAN M' since the distance of subjects is very small.

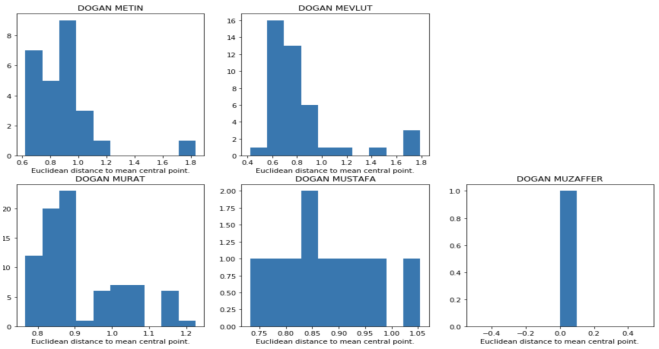

Fig. 2: Histogram of subject distance

C. kNN

We now have 150 papers unlabelled and need to classify them into 6 authors we know. We used word2vec and Google pre-trained archive, and converted the subjects of papers into vectors in 300 dimensions. For each paper, we calculated the averaged keyword vectors. standardization is performed, making the feature mean = 0, std = 1. Then we used these keyword vectors as our features. In figure 3, we used t-SNE to show the samples in 2D plot. We can see that different authors have different subject clusters, so the idea to use subject vectors to do name disambiguation will work to some extent.

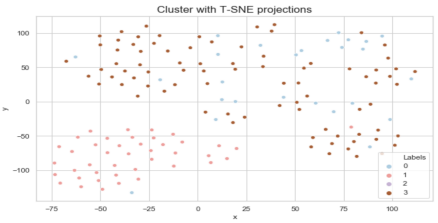

Fig. 3: Visualization of dismissed researchers using t-SNE

Now we have the segments (authors and their mean subjects' vectors), and with KNN we can predict which group an unlabelled paper would will fall into according to its subjects.

To tune the hyperparameter, we used 5-fold cross validation and grid search, and finally found the best parameters were: 'algorithm': 'auto', 'leaf size': 1, 'n neighbors': 5, 'p': 2, 'weights': uniform, with training accuracy 0.84 and test accuracy 0.80.

Then we used the kNN model to predict on unlabelled papers. Since we did not have all DOGAN M* in our dismissal list, papers written by other DOGAN M* may be misclassified into our DOGAN M*. So in next step, we used LOF(local outlier factor) to detect these outlier points.

Local outlier factor is the anomaly score of each sample. It measures the local deviation of density of a given sample with respect to nearest neighbors. It is local in that the anomaly score depends on how isolated the object is in neighborhood. If one sample has lower density to its neighbour, then it is considered as outliers. Label is 1 for an inlier and -1 for an outlier.

And we use jaccard similarity to show the similarity between subjects of labelled data and classified data(unlabelled data). Jaccard similartiy ($J(A,B) = \frac{|A \cap B|}{|A \cup B|}$) calculates the intersection of two sets with a range from 0 to 1.

In our project, higher jaccard similarity means higher similarity of subjects between two sets. Then we compare the jaccard similarity before and after LOF. We chose the parameters with which LOF predicts more -1 (outliers) on papers which contain subject not included in labeled data. For example, DOGAN MEVLUT focuses mainly on physics, atom, optics and nuclear. Before LOF the jaccard similarity is 0.396, and after LOF is 0.442, and we can see that LOF labels papers in automation control and electrical system domain as -1(outlier).

III. Classification

The main objective of classification is to build a model based on a researcher's publications to predict whether he/she would get the sack after the repression of the coup. This classification problem is a bi-classification problem. We plan to prepare the data of both dismissed and undismissed authors and use supervised-learning method to train a model for this task.

A. Pre-processing

As mentioned above, the search strategy of data scraped from Web of Science is to search author's abbreviation name. Therefore, the data we scraped is quite implicit, we have to pre-process it to obtain explicit and necessary data.

In this part, we build 2 datasets—-"Set 1" and "Set 2". In "Set 1", roughly, the main pre-procesing idea is that if one paper has an author with the same full name and organization as one in dismissed/undismissed researcher list, we assign it to this researcher. As a result, from the dismissed researcher list, 2004 dismissed researchers with their paper data were selected into the "Set 1" with Label "1". The undismissed researcher list is created by gathering coauthors of researchers in dismissed list. From undismissed researcher list, we remove researchers who are not from Turkey and the ones with less than 5 publications. Finally 2347 researchers with their paper data were selected into the "Set 1" with Label "0".

However, "Set 1" has the problem that the dismissed and the undismissed can have similar research fields. To preclude the correlations between the research fields, we generate "Set 2". In "Set 2", we replace the undismissed researchers data by collecting 1883 undismissed researchers from the data scraped by searching abbreviated names of the dismissed ones. So the dismissed and undismissed share the same abbreviated names but don't have correlations in terms of research fields. Hence,

we built two datasets, which consist of 4329 and 3865 authors with their paper data respectively, and we use them for the following steps as a comparison.

B. Feature Extraction

Keywords of each publication can indicate which field it belongs to. Thus for each researcher, we accumulate keywords of all his/her collected publications. Then we implement three text representation learning models, Word2vec, Bag-of-words and FastText[1], to translate each keyword into a vector.

Word2vec model has been mentioned in 'NAME DISAMBIGUATION' part, we use the same data as above.

The bag-of-words model is a simplifying representation used in NLP. In our case, we construct the bag-of-words model using the corpus of all key words of undismissed and dismissed author and use the tf-idf (term frequency-inverse document frequency) method to represent the keywords data. The final keywords feature obtained by bag-of-words model is a 4351×44501 sparse matrix whose each row represents keywords feature of one paper and each column represents per distinct key word.

We choose to use FastText instead of Word2vec because it can resolve the errors occuring when words cannot be found in the Word2vec library. FastText model returns a vector of the size 300 for each searched word. Because it is sensitive to word writing forms such as letter case, we first preprocess keywords using capitalization, lowercasing and so forth and then search them into FastText. And we pass those keywords that can't be found after preprocessing. Finally the mean of vectors representing keywords of each researcher is used as the feature and "Set 1" and "Set 2" are represented as 4329×300 and 3865×300 matrices, respectively.

C. Models

The problem we met is a bi-classification problem, so naturally we should try logistic regression model first. For logistic regression model, we apply it to keywords feature got from Word2vec and Bag-of-words respectively. By using 10-fold cross-validation to evaluate the performance of the model, we can tune and select a best penalty parameter C in logistic regression model, which gives us the highest average accuracy of cross-validation.

However, simple linear classification schemes like logistic regression above can sometimes work well but they have their limits. The key to improving such schemes is to well choose features from the original data vector and Neural Network allows us to realize it. In this project, we apply Neural Network method to keywords feature got from Bag-of-word and FastText.

For Bag-of-word feature, since neural Network has very strong representation power such as simple nets with at most two hidden layers are capable of approximating any continuous function arbitrarily closely. We try sequential neural network model with dense layers. The activation function of hidden layers is set to be "rectifier" and that of the output layer is "sigmoid". The loss function is set to be "binary cross entropy" and the optimizer is "rmsprop".

To use features generated by FastText, PCA is performed to reduce the dimension of features before feeding it into Neural Network.

IV. Results

A. Disambiguation

Table 2 shows the result of disambiguation. Column 'name' shows the full name, column 'main subject' shows the main domain the author focuses on(not listing all subjects), column 'similarity' shows the jaccard similarity before and after LOF, and column 'removed subject' shows the subjects predicted -1(outlier) by LOF and thus removed from kNN prediction result.

Name	Main subject	Similarity	Removed subject
DOGAN METIN	biomedicine	0.235 vs 0.222	building, construction
DOGAN MEV-LUT	physics	0.396 vs 0.442	automation control,electrical system
DOGAN MU-RAT	biomedicine	0.421 vs 0.516	atmospheric, meteorology, biodiversity, ecology, sociology

TABLE II: Result of disambiguation

The jaccard similarity of DOGAN METIN dropped after LOF. The reason is that there are only 4 papers predicted to belong to DOGAN METIN, and both share same subject 'life sciences, biomedicine' in their subject list. So even removing one paper will hurt the jaccard similarity. However, in latter two researchers, they have more samples and the effect of removing shraing subject will decrease so we continue on using jaccard similarity.

B. Classification

The effect of principal components kept on the test accuracy of our model integrating PCA and NN with "Set 1" and "Set 2"is demonstrated in Fig 4. We choose the reduced dimension from the range [1, 5, 10, 30, 50, 100, 150, 200, 300]. For "Set 1", test accuracy first increases from near 0.62 to over 0.70 as components rises from 1 to 50. From that point on, the test accuracy gradually declines because the model tends to overfit as more components are kept.

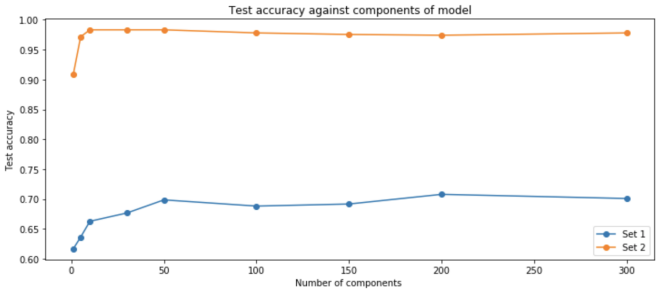

Fig. 4: Test accuracy dealing with "Set 1" and "Set 2"

Fig 4 shows that our model can predict rather well for "Set 2". When only the first principal component is kept, test accuracy is over 0.90. And it reaches the peak of 0.9819 at 30 components. And again test accuracy falls gradually after 30 because of overfitting.

One can see that our model has a significantly better performance when dealing with "Set 2" than "Set 1". The reason for this is that "Set 1" contains dismissed ones and their collaborators, whereas "Set 2" consists of dismissed ones and researchers searched in the database using their abbreviation names. "Set 2" precludes the correlations between the research directions of the dismissed and the undismissed as mentioned above. This leads to the better performance of our model dealing with "Set 2".

Also, we measure how the activation function affects the model's test accuracy. Fig 5 compares six activation functions and shows that 'relu' has the best accuracy.

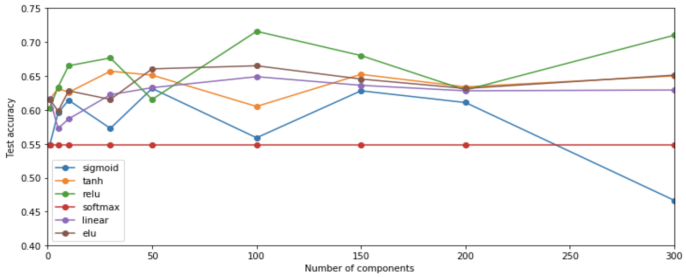

Fig. 5: Test accuracy from 6 activation functions for "Set 1"

As our model predicts not well dealing with "Set 1" compared with dealing with "Set 2", we would show below the enhancement of results when applying Bag-of-words dealing with "Set 1". The results from tuning penalty parameter C in the range [0.01, 0.1, 1, 10, 100, 1000, 10000, 100000] using features given by bag-of-words and Word2vec are shown below. From Fig 6, it can be seen that Word2vec features gets its highest average cross-validation accuracy (0.628) when $C = 10000$ while Bag-of-words feature gets its highest average cross-validation accuracy (0.729) when $C = 1$. Bag-of-words feature works much better than Word2vec feature and its test accuracy of chosen Logistic Regression model ($C = 1$) is 0.709.

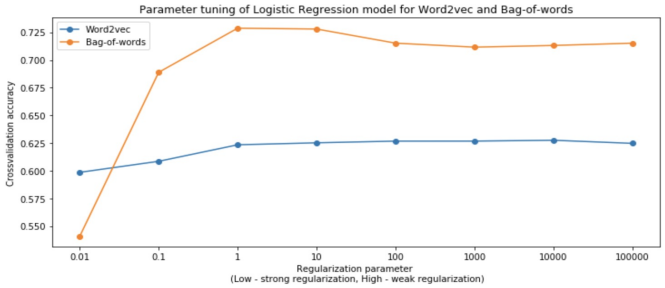

Fig. 6: Parameter tuning of logistic regression model

And we try several activation functions in list ['sigmoid', 'tanh', 'relu', 'softmax', 'linear', 'elu'] and for each activation function, tune epochs from list [10, 15, 20, 25, 30, 35]. The results from adjusting epochs and activation functions are shown in Fig 7:

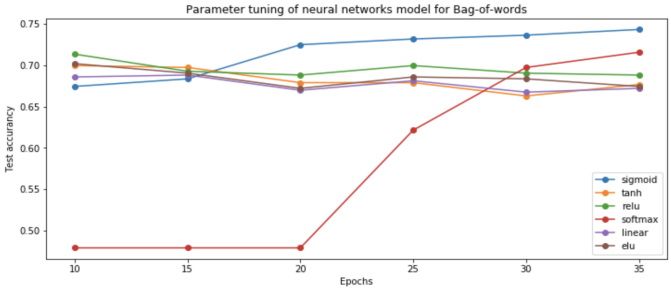

Fig. 7: Parameter tuning of NN model for Bag-of-words

one can see that for Bag-of-words feature, neural networks have the best performance when we use sigmoid activation function and the highest test accuracy reaches 0.747 at epochs = 35.

V. Summary and Discussion

With regard to disambiguation, it's noted that DOGAN MURAT and DOGAN METIN have very close subjects which leads to the misclassification on these two researchers(e.g. in the prediction to DOGAN MURAT we see someone's full name is DOGAN METIN; and same happened in the case of DOGAN METIN). And it's also noted that the misclassification has high kNN probability(eg. the probability of misclassified 'DOGAN METIN' in 'DOGAN MURAT' is 0.8). It reflects a defect of our code: we cannot disambiguate authors in same subject. Actually at first we have tried to use keywords as features to do disambiguation but the result was worse than subject features. We guessed that keywords are more specific and one author have many different keywords, it is harder for us to obtain general feature than using subjects. But, in the case that two authors in same subject, the more specific feature–keyword may be helpful. Regarding the next step to improve disambiguation, we can use keywords as features to disambiguate authors in the same subject.

And we build models to classify the dismissed and undismissed. We introduce three different algorithms to generate features representing keywords and use logistic regression as well as sequential neural networks. It turns out that using features by FastText, neural network can reach an accuracy of 0.98 when the dismissed and undismissed don't have overlap on research fields. And using features by Bag-of-words, neural network can reach an accuracy of 0.747 when classifying the dismissed and their undismissed coauthors. And to improve accuracy in the future, one can integrate citations of a professor in the feature and introduce other techniques such as graphical NNs to investigate the peer effects.

References

[1] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.