FINAL PROJECT REPORT

FINAL REPORT

Doc2Vec Approach for Extracting Knowledge Representation in Research Publications

Student:
Yun Teng

Committee Chair:
Noshir Contractor

Committee Members:
Prof. Huiling Hu
Prof. Alina Lungeanu

Number: NU-CS-2022-13

Doc2Vec Approach for Extracting Knowledge Representation in Research Publications[*]

Abstract

Effective scientific team assembly and collaboration play more and more important roles in innovation for science and technology. As knowledge becomes ever more specialized, analyze team expertise diversity becomes crucial. Thanks to the giant amount of documents and publications that have exploded in the Internet era, we can get the scholar collaboration relation and scientific text easily. However, how to measure team composition and diversity from publication's meta data and the text contained in scientific publications is kind of complex and still an ongoing area to study.

In this report, we propose the definitions of the team's expertise and diversity and illustrates the pipeline for collecting and identifying scientists' prior collaboration networks, conducting text preprocessing, decoding scientists' areas of expertise using text analytics tools such as Doc2Vec approach.

Keywords:

Doc2Vec, Social Network Analysis, Scientific Collaboration

[*] Final Project Report, Department of Computer Science,
Northwestern University, November 22, 2022.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Teams are fundamentally social entities, team's collaboration can be naturally formulated by social network. Social network analysis have been proved to be successful in studies of scientific collaboration (Barabâsi et al., 2002; Wu and Duan, 2015; Abbasi and Altmann, 2011). However, as innovation in science and technology and knowledge becomes ever more specialized, productive scientific team are needed to tackle complex problems requiring insights from multiple domains and effectively collaboration (Jones, 2009). Thanks to the giant amount of documents and publications that have exploded in the Internet era, we can get the scholar collaboration relation and scientific text easily. However, how to measure team composition and diversity from publication's meta data and the text contained in scientific publications is kind of complex and still an ongoing area to study. Also, both textual data and meta data from public datasets may contain lots of typos and miss information, the problems on how to preprocess those data and standardize for scientific used is still needed to be solved.

1.2 Objectives and Contribution

To address aforementioned issues, in this report, we introduced concepts and measures of expertise and diversity for individual team members and proposed an end to end pipeline to leverage researchers' collaboration networks together with the text of researchers' output to identify patterns in the way scientific teams.

Particularly, we used USPTO database Whalen et al. (2020) which contains text vectors for patents based on vector space models to analysis authors' past patents in order to study individual team members' expertise and team's diversity. Here we proposed two concepts, evenness of expertise contributions and divergent ideation on this task and they will be discussed in the following sections. In addition, extending to this task, we used two public dataset, Web of Science and ScienceDirect to generate scientific social networks. Each database has its unique advantages and we combine these two and preprocess the data. In order to utilize both datasets, we design a reasonable matching criteria for these two databases and store the standardized data into MySQL database. Similarly to the approach proposed by Whalen et al., we conducted text analysis and generated text vectors for our further study.

As a conclusion, The main contributions for our work are presented as follows:

- We clear defined the concepts of team members' expertise and team's diversity and conducted analysis on USPTO dataset.
- We combine and matching Web of Science and ScienceDirect

dataset and preprocess them into well structured data into MySQL database.

- we generate scientific social networks from our data and conducted text analysis and generated text vectors for research papers.

2 Methodology

2.1 Patent data analysis

we used USPTO patent data, a rich source of insight for learning more about applied science and technology research groups to study scientific groups' and individuals' expertise and diversity. The data includes the full text of more than 6 million patents granted in the United States since 1976, along with citation data between them, as well as inventors and responsible teams. Thanks to the previous study (Whalen et al., 2020), we have a 300-dimension vector representation for each utility patent documents comprising the description and independent claims text generated by Doc2Vec model (Rehurek and Sojka, 2010). Naturally, we define each individuals' expertise as the mean expertise vector from each individual's prior inventions. Evenness and divergent ideation are proved to play important roles to team outcomes (Bales and Strodtbeck, 1951; Woolley et al., 2010; Runco, 2010). For each published patent, we have a scientific team where each team member has there own expertise vector. We define the evenness of expertise contributions as the degree to which team members' prior areas of expertise are represented to a similar extent in the team's products. We also define divergent ideation as the degree to which the knowledge artifacts produced by the team reflect areas of expertise that differ from members' prior expertise. To measure individuals' expertise difference, we can formulate this task as a vector distance measures and it can be used to provide insight into how similar or dissimilar each individuals' expertise differ from each other in the same team or prior expertise of each individual own. Cosine similarity is a commonly-used vector distance measure. It is defined as:

$$Similarity(\mathcal{A}, \mathcal{B}) = \frac{\mathcal{A} \cdot \mathcal{B}}{\|\mathcal{A}\| \, \|\mathcal{B}\|} \tag{2.1}$$

where $\mathcal{A} \cdot \mathcal{B}$ is the dot product between two vectors $\mathcal{A}, \mathcal{B}$ and $\|\mathcal{A}\|, \|\mathcal{B}\|$ are the norms of vectors A and B respectively. Similarly, cosine distance is measured as:

$$Distance(\mathcal{A}, \mathcal{B}) = 1 - Similarity(\mathcal{A}, \mathcal{B}) \tag{2.2}$$

We demonstrated a simplified example of how this approach works in Figure 2.1.

So first we identify all co-inventors of a patent P: I_1, I_2, I_3. For each inventors, we identify all prior inventions for each inventor. For this example, Inventor I_1 has three prior patents, $P1_{I1}, P2_{I1}, P3_{I_1,I_2}$, inventor I_2 has four prior patents, $P3_{I_1,I_2}, P4_I2$, $P5_I2, P6_{I2}$, and inventor I_3 has two prior patents, $P7_{I3}, P8_{I3}$. Next, we can simply compute inventors' expertise as the mean expertise vector from each inventor's prior inventions: E_{I1}, E_{I2}, E_{I3}. Now we define team diversity (evenness of expertise contributions) as the distance between each mean expertise point and taking the maximum distance of the

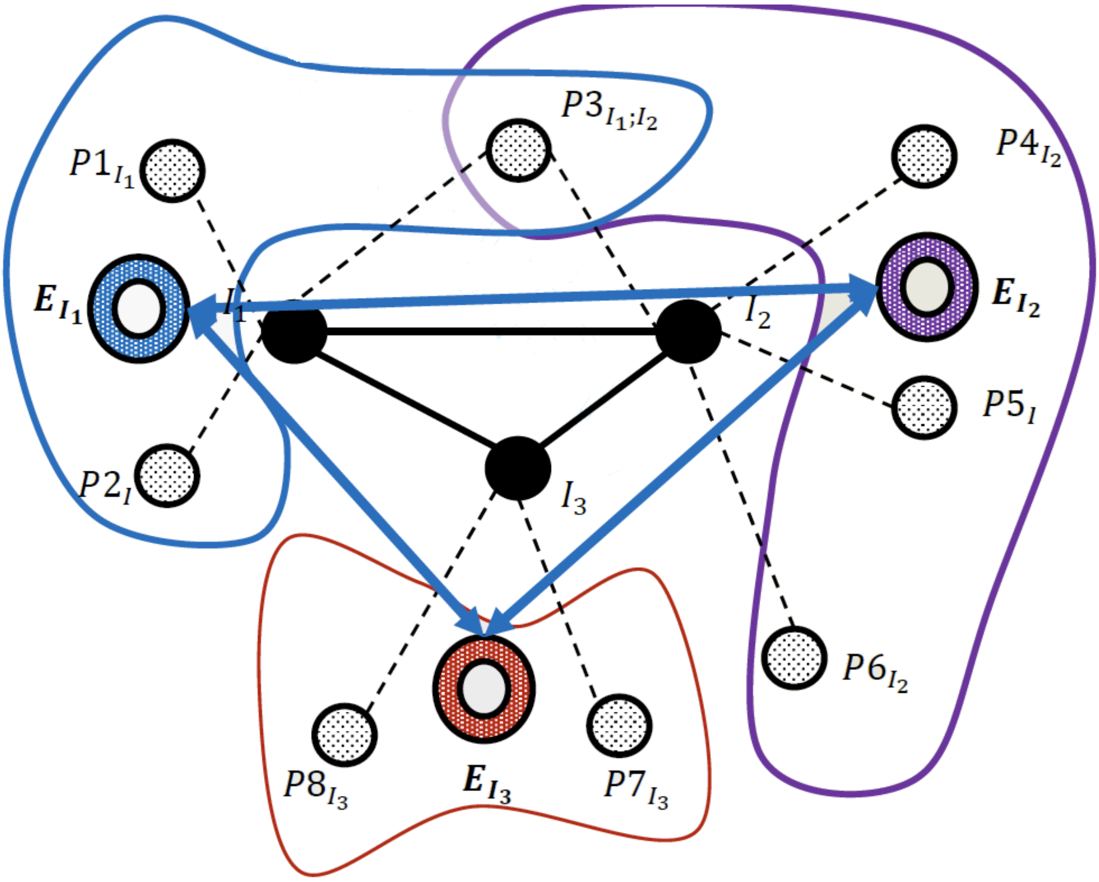

Figure 2.1: Computing team evenness of expertise contributions and divergent ideation

three. Also, we compute the distance between each inventors' expertise to the patent P to represents the divergent ideation.

Here is the simplify code for calculate the divergent ideation from our database:

```
""# caulate the divergent ideation
for inventor_id in tqdm(inventor_key_dict):
    patents = inventor_key_dict[inventor_id]
    valid_patents = []
    ideation = []
    write_distance = []
    for patent in patents:
        if patent not in patent_time_dict or patent not in patent_vec_dict:
            print('No date or vec data for inventor_id {0}'.format([inventor_id]))
        else :
            valid_patents.append((patent, patent_time_dict[patent]))

    valid_patents = sorted(valid_patents, key=lambda x: x[1])

    length_valid = len(valid_patents)
    if length_valid >= 2:
        for i in range(1, len(valid_patents)):
            if ideation:
                previous = ideation[-1][-1]
            else:
                previous = np.zeros(300)
            idea = (previous* (i - 1) + asarray(ast.literal_eval(patent_vec_dict[valid_patents[i - 1][0]])))/i
            similarity = cos_sim(idea, asarray(ast.literal_eval(patent_vec_dict[valid_patents[i][0]])))
            distance = (1 - similarity)/2
            ideation.append( (valid_patents[i][0], valid_patents[i][1], idea) )
            write_distance.append((valid_patents[i][0], valid_patents[i][1], distance))
        write_distance = [inventor_id] + write_distance
```

And here is the simplify code for calculate the eveness:

```
""# caulate eveness
for patent_id in tqdm(patent_vec_dict):
    if patent_id not in patent_key_dict:
        print('Key error for patent_id {0}'.format([patent_id]))
        continue

    if len(patent_key_dict[patent_id]) > 1:
        date = patent_time_dict[patent_id]
        patent_vec = patent_vec_dict[patent_id]
        similarity_list = []
        valid_inventor_count = 0

        for inventor in patent_key_dict[patent_id]:
            valid_patents = []
            patents = inventor_key_dict[inventor]
            for patent in patents:
                if patent not in patent_time_dict:
                    print('No date data for patent_id {0}'.format([patent_id]))
                elif patent_time_dict[patent] < date and patent in patent_vec_dict:
                    valid_patents.append(patent_vec_dict[patent])
            if valid_patents == []:
                similarity = -1
            else:
                valid_inventor_count += 1
                valid_patents_vec = str2vec(valid_patents)
                inventor_vec = np.sum(valid_patents_vec,axis=0)/len(valid_patents_vec)
                similarity = cos_sim(inventor_vec, patent_vec)
            similarity_list.append((1-similarity)/2)
        writer.writerow([patent_id, np.min(similarity_list), np.max(similarity_list), np.mean(similarity_list), np.std(similarity_list), len(patent_key_dict[patent_id]), valid_inventor_count])
```

2.2 Research paper analysis

Extended to previous patent analysis, we want to generate our own database on research papers and perform the similar analysis. We used two public database, Web of Science and ScienceDirect and perform a designed matching strategy for taking advantage of both databases. Finally we trained paper's text vectors for future analysis.

2.2.1 Original database

The Web of Science includes information on authorship, location, institution, citations, journals and keywords for all areas of science and engineering, social sciences, and humanities, with most types of data available from 1945 to today. The Web of Science database includes the Digital Author Identification System (DAIS) number, a unique internal ID used to disambiguate authors.

The Elsevier ScienceDirect database provides the XML version of the full text of more than 6.5M articles published across 255 different fields in more than 2,500 journals since 1823 when it had 165 publications. This will allow us to reveal researchers' expertise based on the text of their publications and the text used by other publications when citing their papers.

2.2.2 Matching strategy

We use this dual database approach because of the distinct qualities of the data available to us in each. We have full text of articles from ScienceDirect, but these do not contain unique identifiers for citations or authorship information. So, we match the ScienceDirect articles with their Web of Science entries and extract authorship and citations information from the Web of Science. For matching two databases, we consider using author name, paper title and publication year as our matching entries. After exploring, because author name may unexpected abbreviated to the initial for first and last name in both databases, we decide to use paper title and publication year as our matching criteria. At the first stage, we used exact match on the whole text of paper names together with publication year. However, through this criteria, some paper may mismatched in some special cases showing below. For each cases, the upper one represents the publication name in the ScienceDirect database and the lower one represents the publication name in the Web of Science database. The difference bewteen two publication name is highlight with red.

- Sample case 1: dash
  - “Discussion of the effective stiffnesses in: Ye, Berdichevsky, and Yu [Int. J. Solids Struct. 51 (2014) 2073 -2083]"
  - “Discussion of the effective stiffnesses in: Ye, Berdichevsky, and Yu [Int. J. Solids Struct. 51 (2014) 2073–2083]”
- Sample case 2: Greek letter
  - Phenolic profiles of 20 Canadian lentil cultivars and their contribution to antioxidant activity and inhibitory effects on α-glucosidase and pancreatic lipase
  - Phenolic profiles of 20 Canadian lentil cultivars and their contribution to antioxidant activity and inhibitory effects on alpha-glucosidase and pancreatic lipase
- Sample case 3: parenthesis
  - Genotoxic effect of dimethylarsinic acid and the influence of co-exposure to titanium nanodioxide (nTiO(2)) in Laeonereis culveri (Annelida, Polychaeta)
  - Genotoxic effect of dimethylarsinic acid and the influence of co-exposure to titanium nanodioxide (nTiO2) in Laeonereis

culveri (Annelida, Polychaeta)

- Sample case 4: missing a part of the title
  - Long-Term Outcomes of the FORMA Transcatheter Tricuspid Valve Repair System for the Treatment of Severe Tricuspid Regurgitation Insights From the First-in-Human Experience
  - Long-Term Outcomes of the FORMA Transcatheter Tricuspid Valve Repair System for the Treatment of Severe Tricuspid Regurgitation

To resolve these cases, we do the preprocessing on puclication titles to remove all non-alphabet characters (e.g., numbers, parentheses, dashes, spaces) and replace all Greek and German letters into English letters. Also, we lower all the alphabet characters for both databases' publication name and takes first 100 characters for matching. Sample cases above have been well handled by our preprocessing. We used author name to manually check our matching result, only 8000 data points got different authors, however, most of them are misspelling, first name failed to spilt, character different (Thiaener vs Thiäner), and unexpected abbreviated to the initial. Overall, preprocessing is quite useful.

2.2.3 Document to vectors

With access to publication text, contemporary natural language processing techniques allow us to move beyond metadata measures of expertise, interdisciplinarity and impact. Doing so requires first transforming article text into a vector space model, which can then be used to measure the distance between articles or patents in n-dimensional space. We know that all the analysis proposed in Section 2.1 rely extensively on the use of vector space models (VSM) to represent text in n-dimensional space. For this task, our goal is to train a VSM to generate vectors from research paper to model knowledge space. Plenty of natural language processing techniques allows for the text of a researcher's output to be situated in a modeled n-dimensional space, which subsequently enables distance measures between it. Models using more dimensions are able to represent more fine-grained distinction between topics such as the distinction between quantitative and qualitative sociology. Documents can be represented as vectors inside the modeled space once a Doc2Vec model has been trained on a corpus. We may further analysis team's diversity and individual's expertise by computing the vector distance between them.

To demonstrate a simplified example of how this approach works, imagine three research outputs: Article A is an epidemiological article about the spread of influenza across social network ties; Article B is a social science article about how hashtags spread on online social networks; and Article C is a gender studies article about non-binary gender politics. In a simple 10-dimension model, we might imagine their representation is as below in Table 2.1. These values represent the degree to which each of these articles is related to each dimension within the 10-dimension model.

Dimension	1	2	3	4	5	6	7	8	9	10
Article A	0.1	0.2	0.4	0.1	0.3	0.5	0.9	0.1	0.9	0.1
Article B	0.6	0.2	0.5	0.1	0.3	0.8	0.9	0.2	0.9	0.3
Article C	0.3	0.2	0.4	0.7	0.7	0.5	0.8	0.1	0.2	0.1

Table 2.1: Simplified example of vectors presentations for articles

2.2.4 Train the model

The literature on natural language processing has created a number of vector space models and used those models to analyze data from diverse substantive domains, proving the value of those models for topic modeling and information retrieval (Rehurek and Sojka, 2010; Goldberg and Levy, 2014; Lau and Baldwin, 2016; Kim et al., 2019). Recent developments in machine learning have enabled highly-performative vector space models such as Word2Vec and Doc2Vec Goldberg and Levy (2014); Lau and Baldwin (2016).

The Doc2Vec model is an extension of the Word2Vec model. In Word2Vec, we train a neural network to anticipate the following word from a prior word sequence (context). The model in Doc2Vec learns representations for both paragraphs and words. The position of the document inside the modeled vector space can finally be represented by the representation of paragraphs in vector space. This vector space representation has been usefully proved in studies for practical applications like sentiment analysis and information retrieval, and it has a number of benefits over other vector space models like TF-IDF, LSA, LDA, etc. Bilgin and Şentürk (2017); Kim et al. (2019).

Practically, we used online tool Gensim [1]. To train the model, we could train the model on all publications and obtain the vectors which is the most straightforward and accurate approach. However, our database scacle is large which makes this approach less piratical. We can also obtain a pretrained model online and infer all the publications by the pretrained model. There are some potential options for the pretrained models like model training on English Wikipedia or Associated Press News [2] and model training on movie data [3]. However, the corpus are too irrelevant from our research papers database which are not appropriate to directly use. To solve this challenge, we split our dataset 10% of publications for each year to train the model, and inference the remain papers.

Here is the simplify code for training the model on our corpus:

```
""# create and train model
doc_iterator = DocIterator()
    model = gensim.models.Doc2Vec(
        documents=doc_iterator,
        workers=n_cpus,
        vector_size=300,
        epochs = 12
        )

    model.delete_temporary_training_data(keep_doctags_vectors = True
    ,
                                        keep_inference = True)
    model.save('sd_2014_12e.model')
    model = gensim.models.Doc2Vec.load('sd_2014_12e_test.model')
    write_vectors()
```

[1] We use the gensim tool obtained by https://radimrehurek.com/gensim/

[2] https://github.com/jhlau/doc2vec

[3] https://iboxshare.com/sindbach/doc2vec_pymongo

3 Conclusions and future work

Effective scientific team assembly and collaboration are important for scientific outputs. In this report, we proposes concepts of evenness and divergent ideation to meature teams' diversity and individuals' expertise and how to compute those measures. We also illustrate the procedures of generating our own research papers vector databases and methods to preprocess data and train the models. In the future, we could look deeper in to our measures matrix and explore the relatedness with our measures to scientific success (e.g. citation rates) using predict analysis like regression analysis. Also, as innovation of text analytic approach, we could update the techniques for training our model for more accurate outputs.

Bibliography

Abbasi, A. and Altmann, J. (2011). On the correlation between research performance and social network analysis measures applied to research collaboration networks. In *2011 44th Hawaii international conference on system sciences*, pages 1–10. IEEE.

Bales, R. F. and Strodtbeck, F. L. (1951). Phases in group problem-solving. *The Journal of Abnormal and Social Psychology*, 46(4):485.

Barabâsi, A.-L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., and Vicsek, T. (2002). Evolution of the social network of scientific collaborations. *Physica A: Statistical mechanics and its applications*, 311(3-4):590–614.

Bilgin, M. and Şentürk, İ. F. (2017). Sentiment analysis on twitter data with semi-supervised doc2vec. In *2017 international conference on computer science and engineering (UBMK)*, pages 661–666. Ieee.

Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Jones, B. F. (2009). The burden of knowledge and the "death of the renaissance man": Is innovation getting harder? *The Review of Economic Studies*, 76(1):283–317.

Kim, D., Seo, D., Cho, S., and Kang, P. (2019). Multi-co-training for document classification using various document representations: Tf–idf, lda, and doc2vec. *Information Sciences*, 477:15–29.

Lau, J. H. and Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.

Rehurek, R. and Sojka, P. (2010). Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*. Citeseer.

Runco, M. A. (2010). Divergent thinking, creativity, and ideation.

Whalen, R., Lungeanu, A., DeChurch, L., and Contractor, N. (2020). Patent similarity data and innovation metrics. *Journal of Empirical Legal Studies*, 17(3):615–639.

Woolley, A. W., Chabris, C. F., Pentland, A., Hashmi, N., and Malone, T. W. (2010). Evidence for a collective intelligence factor in the performance of human groups. *science*, 330(6004):686–688.

Wu, Y. and Duan, Z. (2015). Social network analysis of international scientific collaboration on psychiatry research. *International Journal of Mental Health Systems*, 9(1):1–10.