Data Science In Practice Assignment 3

Rita Abani 19244

Anushka Shreyam 20050

Topic : Decoding Author Name Disambiguation through the lens of Data Science manifested in the forms of Graph Neural Networks, Network Topology and Deep Learning

Introduction

Author name disambiguation is of pertinent interest to the Data Science Community owing to the circumstantial nature of an individual's writing/ publishing track record. Individuals may publish under multiple names for a variety of reasons including different transliteration, misspelling, name change due to marriage, or the use of nicknames or middle names and initials.

According to Morrison et al. motivations for disambiguating individuals include identifying inventors from patents. In the work ' *ArnetMiner: extraction and mining of academic social networks* , according to the authors, 'Name disambiguation is also a cornerstone in author-centric academic search and mining systems, such as ArnetMiner

An editor may apply the process to scholarly documents where the goal is to find all mentions of the same author and cluster them together. Authors of scholarly documents often share names which makes it hard to distinguish each author's work. Hence, author name disambiguation aims to find all publications that belong to a given author and distinguish them from publications of other authors who share the same name.

In this assignment, the authors aim to explore the challenges, motivation and implications of author name disambiguation.
The first case study involves Data Science mingled with representation learning conveyed through Graph Neural Networks, a mathematically emerging paradigm

The second case study involves approaching the problem from a symbolic and semantic perspective by making use of Neural Networks.

Case Study 1 : A Graph- Based Author Name Disambiguation Method and Analysis via Information theory by Ma et al.[1]

Context

The purpose of an author name disambiguation task is to divide documents related to an author name reference into several parts, each associated with a real-life individual. In existing methods, either attributes of documents or relationships between documents and co-authors are used. A feature extraction approach based on attributes

results in inflexibility of models, while a solution based on relationship graph networks ignores features. In this paper, Ma et al. present a novel name disambiguation model which incorporates attributes and relationships based on representation learning. The model is demonstrated to be effective on a public dataset, and experimental results confirm that the solution is superior to several state-of-the-art graph-based approaches. The model also increases the interpretability of the deployed method through information theory and shows that the analysis could be helpful for model selection and training progress.

Key words : name disambiguation, graph neural network, clustering analysis, mutual information

Approach and pedagogy :

By completing the author name disambiguation exercise, the authors try to answer two different types of questions : to determine which articles are written by an author and which author has written which article (reverse mapping). Author name disambiguation becomes more challenging when dealing with scholars not from western countries, as people with different names share the same spelling. For example, there are more than 300 thousand people named “Wei Zhang” in China as pointed out by the authors.

- **Architecture** :

The method proposed involves generating document embedding vectors without labeled data and utilizing the true number of clusters for clustering. It makes use of the relationship between authors as well as the relationship between papers. Incorporating a word2vec model, the model uses basic attributes for feature extraction namely: title, author, organization and constructs document representation vectors.
According to the official TensorFlow documentation, word2vec is not a singular algorithm, rather, it is a family of model architectures and optimizations that can be used to learn word embeddings from large datasets. Embeddings learned through word2vec have proven to be successful on a variety of downstream natural language processing tasks. Word2vec is namely deployed in the following 2 scenarios :

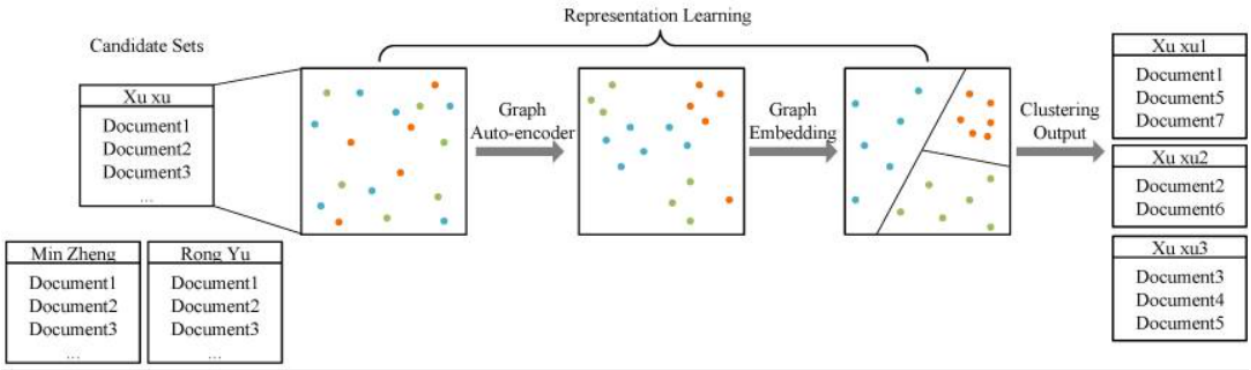
- **Continuous bag-of-words model**: predicts the middle word based on surrounding context words. The context consists of a few words before and after the current (middle) word. This architecture is called a bag-of-words model as the order of words in the context is not important.
- **Continuous skip-gram model**: predicts words within a certain range before and after the current word in the same sentence.

A paper-paper network based on feature similarity was built using a graph auto-encoder (GAE), and the graph topology information of the co-author relationship is used to inform the graph embedding model.

The outcomes of the model's disambiguation are assessed using a larger dataset sampled from AMiner . According to experimental findings, the solution outperforms a number of cutting-edge graph-based techniques, including Zhang and Yao, Zhang et al. and GHOST The approach deployed here, which uses an unsupervised learning approach to find the document embedding vectors, is even more effective than the semi-supervised approach.

By monitoring the changes in mutual information, the interpretability of the model is increased via principles of information theory. DImensionality reduction based visualization has been used further to compare the model with existing worlds and assess the role of individual components.

- Framework:



The vibrant dots in Figure 1 reflect representations of documents. These representation vectors are calculated using a word2vec model. These vectors are then fine-tuned using a variational Graph Auto-Encoder and Graph Embedding model. After representation learning, the documents are divided into various clusters using hierarchical agglomerative clustering.

This paper deploys the **Continuous Bag-of-Words (CBOW)** to learn the representation of words.
Given a series of training words w1,w2,…,wT, CBOW learns word representation based on the co-occurrence of words within the size of a predefined context window in the training corpus. The model is based on the idea that the probability of occurrence of words around a word can predict what the word is. It maximizes the co-occurrence probability between words that appear in the predefined context window. The objective is to maximize the probability represented through the mathematical formulation given below :

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p\left(w_t|w_{t+j}\right)$$

Where c is the context window. The loss function is optimized by a neural network. The training words are represented by the weight matrix after training.

A particular document's dn could be written as "f1,f2,...,fM," where "f" stands for the stemmed words "title," "co-authors," "organization," and "venue." Each feature f is represented by one hot vector. Here, each document representation is condensed into a low-dimensional vector using the CBOW paradigm. The low-dimensional representation of feature fm is represented by the m-th weight of the neural network weight matrix after training. The formula dn=Mi=1ifi could be used to calculate every document representation dn. To lessen the importance of some unnecessary stemmed words, such as prepositions, I is the inverted document frequency of fi. Using the co-occurrence probability of the attributes in each article, Di=d1,d2,...,dN captures the similarity between these publications.

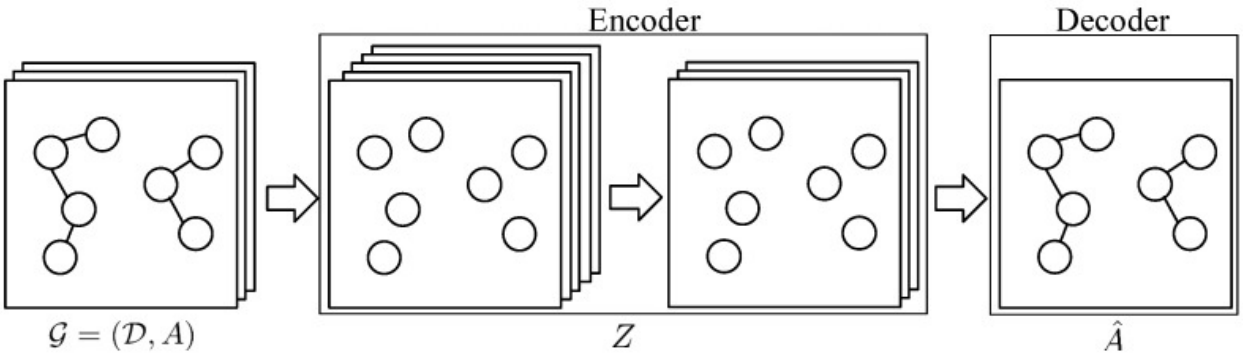


Figure 2 : Variational Graph Auto-Encoder is used in the model to improve the generalization ability of the model. Let X=[dT1,dT2,…] be the representation matrix of documents associated with an author name. The encoder is a two-layer graph convolutional network.
D={d1,d2…,dn} is the document representation output Di. The document embedding vectors could represent graph nodes and the adjacency matrix A here represent edges between these nodes.

The adjacency matrix A is constructed by two approaches: one is calculating affiliation similarity between two documents, and the other one is using a collection of high-precision features.

The latent matrix output of encoder Z=[zT1,zT2,…] is taken as new embedding vectors of these documents. By knowing the relationship of existing nodes, the new matrix is a higher-level expression and has the predictability of new node pairs.

Evaluation and experimental results :

The same hyperparameters have been used for each dataset. In the CBOW model, the dimension of document representation vectors is set to 100 and the context window is set to 5. In the variational Graph Auto-Encoder, the threshold of inverse document frequency is 25, the output dimension of the first layer in graph convolution network is 200, the output dimension of the second layer is set to 100, the learning rate is 0.01 and the model trained for 200 epochs. In the graph network embedding model, the learning rate is 0.05 and the regularization parameter is 0.01. The model achieves the best performance comparing with other graph-based methods in F1-score.

Graph Auto-Encoder and Graph Embedding improve the performance by 6.41% and 4.83% respectively.

Table 4

Clustering Results of Each Component.

	Prec	Rec	F1
Feature Embedding	72.29	50.14	59.21
Graphh Auto-Encoder	75.53	58.01	65.62
Graph Embedding	77.71	54.46	64.04
Overall	78.10	67.47	72.40

Case Study 2 : Deep Learning Approach for Author Disambiguation using Bibliographic Data by Han et al.[11]

The author proposed a novel approach to link author names to their real-world entities by relying on their co-authorship pattern and area of research. The supervised deep learning model identifies an author by capturing his/her relationship with his/her co-authors and area of research, which is represented by the titles and sources of the target author's publications. These attributes are encoded by their semantic and symbolic representations. The extensive experiments have proved the capability of the approach to distinguish between authors sharing the same name and recognize authors with differ- ent name variations. Bib2Auth has shown good performance on a relatively large dataset, which qualifies it to be directly integrated into bibliographic indices.

KEYWORDS

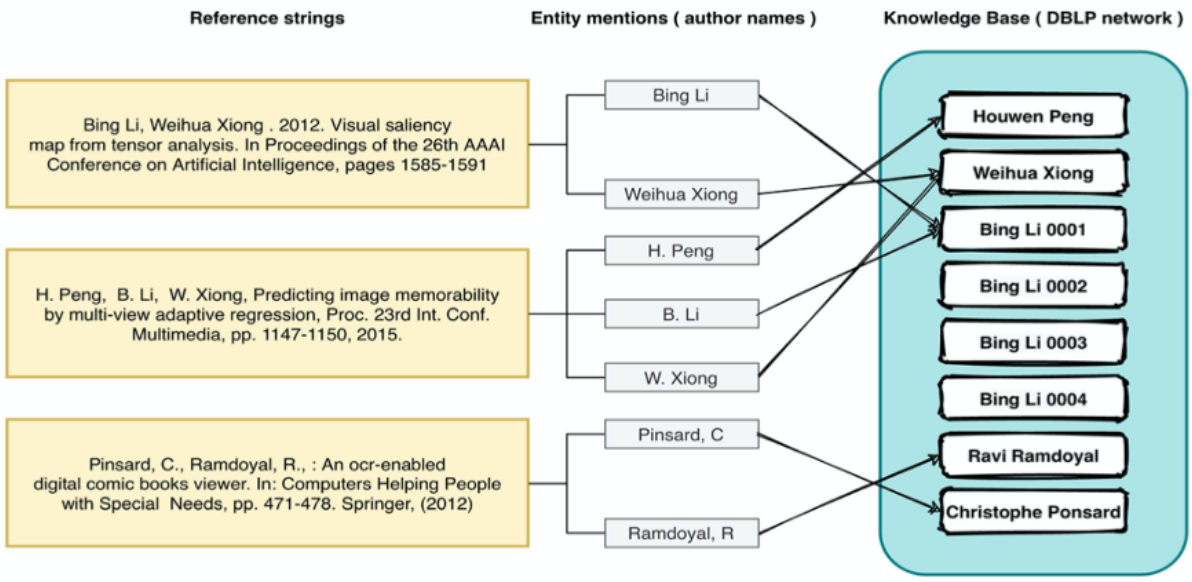
author name disambiguation, neural networks, classification

Main Contributions:

- Proposed a novel approach for author name disambiguation using symbolic and semantic representation.
- proposed a challenging dataset for author name disambiguation.
- The experimental results on the challenging dataset demonstrate the effectiveness of our Bib2Auth to disambiguate author names.

BIB2AUTH :

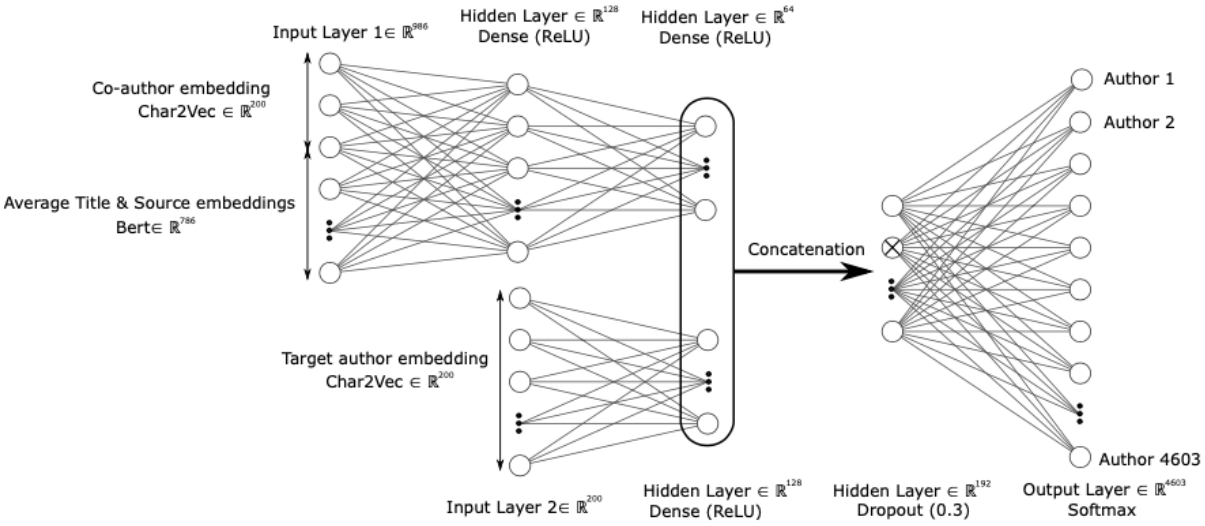
In this paper, the process of author name disambiguation is defined formally as: Let R = {$r1$, $r2$, · · · , rM } be a set of M reference strings, A = {$a1$,$a2$,··· ,aN } be a set of N author name mentions and E = {$e1$, $e2$, · · · , eK } be a set of K unique author entities in DBLP. The goal of Bib2Auth is to map each name mention (i.e. author name) $an \in A$ in the reference $rm \in R$ to the corresponding unique entity $ek \in E$ with DBLP identifier.



Above figure is an illustration for the task of linking a name mentioned in the reference string with the corresponding DBLP author entity.

Model Architecture:

This method is based on the two input layers Neural Network Model. The first input layer represents the concatenation of the co-author embedding and the content embedding. The latter is the average embedding of title and source. The second input layer represents the embedding of the target author. For author and co-author, the embedding is of length 200 and is generated using Char2Vec [3], which is a recurrent neural network that provides a symbolic representation of the given word. For title and source, the embedding is of length 786 and is generated using BERT [5] which provides a vector representation of words w.r.t their context in the sentence. The goal of separating the two inputs is to overcome the sparseness of the content embedding and force the model to emphasize more on target author representation. This Bib2Auth model having an output layer of length 4603 corresponds to the number of authors in our dataset. Here all hidden layers possess ReLU activation function, while the output layer possesses a Softmax activation function. Since the model needs to classify thousands of classes, each of which is represented with very few samples, 30% of the units in the last hidden layer are dropped out during training to avoid overfitting. Therefore, each class (i.e., author) is weighted according to the number of its samples (i.e., publications). The model is trained using adam optimizer and the sparse categorical cross-entropy loss function.



Above figure shows the actual visual architecture of the Bib2Auth model.

Considerable Embedding:

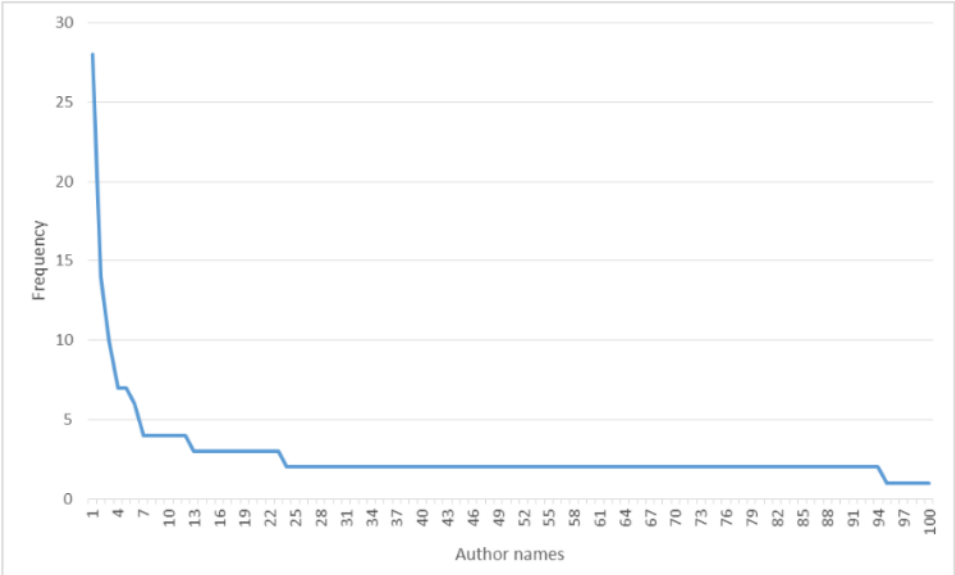
- ➔ Author Name Embedding : used the Chars2vec model, which is implemented using Keras based on TensorFlow.
- ➔ Source Embedding : uses the Venue API returns data in an HTML format upon which we used the beautiful soup4 to retrieve the journal names.
- ➔ Title Embedding : uses pre-trained BERT model to generate sentence embedding for the titles.

By further doing model tuning, Bib2Auth model fine-tunes the parameters to predict the appropriate target author.

Although Bib2Auth stops the training process when it reaches a minimum validation loss, the model obtained at the end of the training may not give the best accuracy on validation data. To account for this, Keras provides an additional callback called ModelCheck point. With this setup, the model updates the weights only when it observes better validation accuracy compared to earlier epochs. Eventually, we end up persisting with the best state of the model with respect to the best validation accuracy.

Experimental validation of Bib2Auth Model:

- ***Datasets***: Using the DBLP bibliographic repository5 as a source of data to train, validate and test Bib2Auth. Also, the authors in DBLP who share the same name have a suffix number to differentiate them. For instance, the authors with the same name ’Bing Li’ are given suffixes such as ’Bing Li 0001’, ’Bing Li 0002’. Figure 3 shows the frequency of the first 100 authors sharing the same name (first and last names) in the dataset used by Bib2Auth.
    - The Frequency of author names in the used datasets:



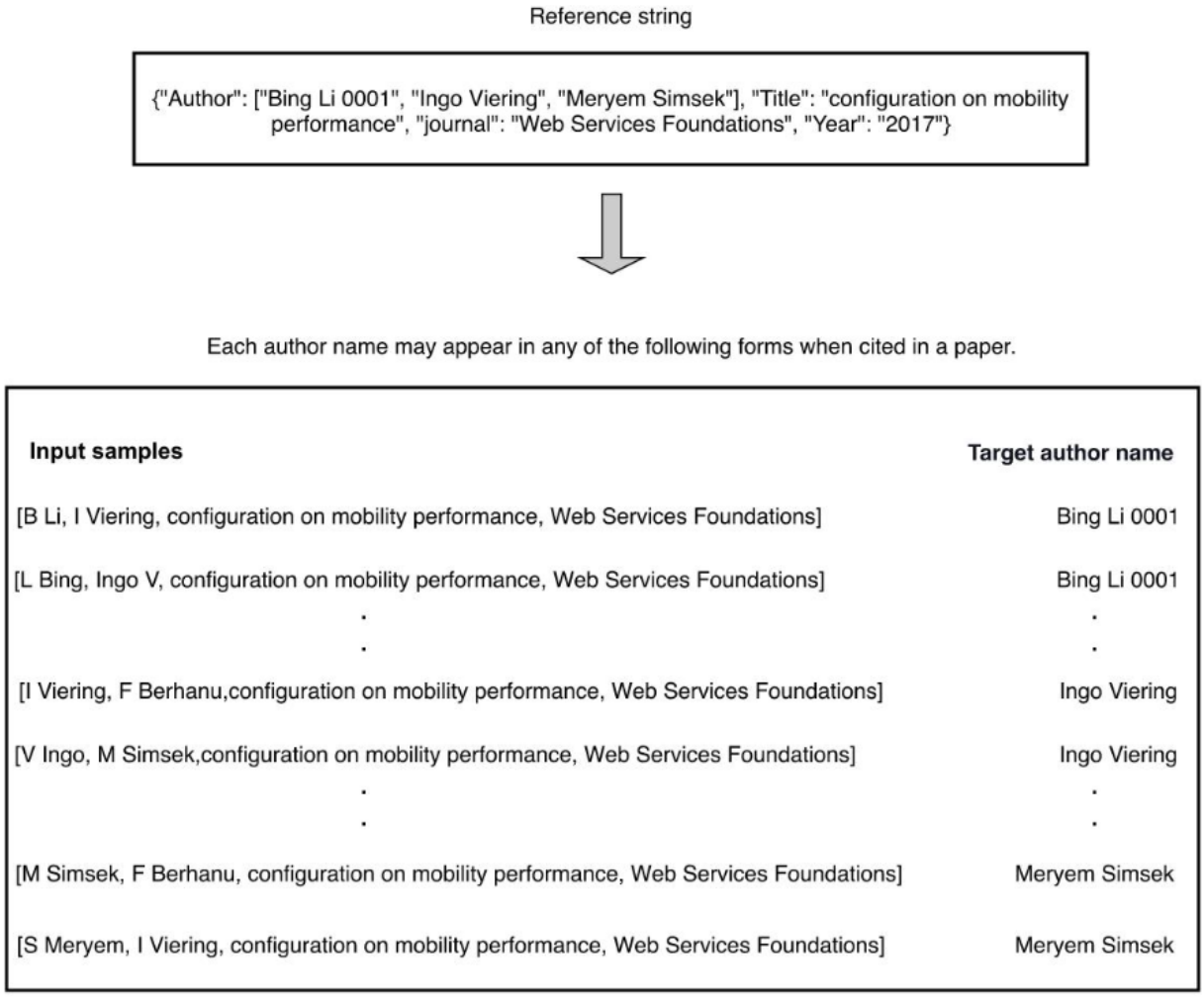
- ***Training the model*** : Since training the model on 4.4 million records is very time-consuming, we used a sample dataset of 21802 records which are randomly sampled from the DBLP repository. However, it has been ensured that the dataset contains records from different authors sharing the same first and last name. Since authors publish papers with different co-authors, it is challenging to sample a dataset while ensuring that all or at least some publications from all co-authors are present in the training set.
- ***Corner Cases*** : As each record may have one or more authors, it is not possible to feed input samples of variable length to our model. Therefore, we limit the number of authors in a sample to two. One acts as a target author and the other as a co-author. Thus, we generate all possible name combinations of author-co-author pairs from each record. This means that each record is represented by multiple input samples with the same title and source, but with different pairs of author names. Note that all co-authors are considered as target authors in the generation of input samples. To account for possible citation styles for author names, we further enrich the input sample by including all name variations for author and co-author. If the first and last name of the target author and co-author consist of one token, six pairs of variations are generated. Consequently, each sample is a tuple consisting of an author name variant, a co-author name variant, a title, and a journal. Figure 4 illustrates an example of generating multiple samples from one bibliographic record to capture possible name variants of authors. With this strategy, we assume that the model can capture all possible variations of author names. The class label of each such

sample is the original author from the DBLP corresponding to the author name variant.

★ ***Statistical Details of the used Datasets:***

	Combination	Records	# unique authors
Training set	45060(~ 66%)	2534	$\|C\| = 4603$
Validation set	11584(~ 17%)	10952	$2537 \in C$
Testing set	11584(~ 17%)	10850	$2576 \in C$

★ ***Examples of Generating Input Samples:***



For above records, only combinations of co-authors that were present in the training process are considered for validation and testing. The reason for preparing the data in

such a way is to ensure that records from training are not used either in validation or testing. Specifically, each of the three sets has unique publications (i.e., titles).

- ***Results*** :

  The obtained results of Bib2Auth on a test dataset for each pair of co-authors.

	Macro	Micro
Precision	0.989	0.975
Recall	0991	0.975
F1-Score	988	0.975

- The obtained results of Bib2Auth on a test dataset for each pair of co-authors, considering only full names. (represents the Macro and Micro average of precision, recall, and F1-score)

	Macro	Micro
Precision	0.993	0.984
Recall	0996	0.984
F1-Score	994	0.984

Above results demonstrate the ability of Bib2Auth to distinguish between authors sharing the same name when only a co-author, title, and source of publication are given. The obtained results also prove the effectiveness of the model in handling differ- ent name variants.

Evaluated model on train, validation and test data with a split ratio of 66% : 17% : 17%. The training and validation loss is monitored continuously over the optimal number of epochs. We run the model on each combination of the test set.

- ***Conclusion*** :

In this paper, the challenges associated with entity linking for author names in citation strings. Further proposed Bib2Auth, a novel framework for author name disambiguation. Bib2Auth is a supervised deep learning model which leverages different reference attributes such as co-authors, title, and journal to perform collective disambiguation. Experimental results have shown that Bib2Auth achieves very promising and satisfactory results on a challenging dataset, which is full of authors sharing the same names.

Despite achieving high accuracy, there are several avenues for improving and enhancing Bib2Auth. First, the authors of the paper plan to train the model on the entire DBLP repository by following a hierarchical classification so that authors with completely dissimilar names can be pre-filtered before training. This allows training on a large number of bibliographic records as well as on a high number of authors. Consequently, the time required for training and testing will be reduced.

Inference :

The versatility of data science to infer trends in mathematical, statistical and analytical problems applied to solve real world problems like author name disambiguation make it a robust tool to study, propagate and imbibe in other models that are based on conventional brute force approaches.

References

1 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7516896/
2 https://www.tensorflow.org/tutorials/text/word2vec
3.https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7516896/#sec4-entropy-22-00416
4.https://arxiv.org/pdf/2107.04382.pdf
5.https://icml.cc/virtual/2022/events/workshop
6.https://direct.mit.edu/qss/article/1/4/1510/96105/Author-name-disambiguation-of-bibliometric-data-A
7.https://analyticsindiamag.com/a-comprehensive-guide-to-representation-learning-for-beginners/
8.https://towardsdatascience.com/a-gentle-introduction-to-graph-neural-network-basics-deepwalk-and-graphsage-db5d540d50b3