Tracking the seminal work creator in big scholarly networks

Mengxin Liu
515021910215

Wencheng Tang
515021910424

***Abstract*—Tracking the seminal work creator in big scholarly networks can help researchers to have a deep understanding about their field of interest. We are trying to applying the source locating methods in the scholarly networks to solve this problem. In order to ensure the reliability of our results, in this paper we use a real scholarly dataset from Acemap datasets, which provides more than 1.5 million papers in the computer science field. In order to divide the field, we use the topic information from Acemap datasets and text cluster methods. Then, we also proposed a classifier to detect the field of new paper using deep neural networks. Finally, we define the source in the fields and Analysis the results. Whether a paper in the field of computer is in the dataset or not, our model will provided the seminal work creator of the related field.**

I. Introduction

With the rapid advancement of science and technology, identifying seminal work creator in big scholarly networks has become increasingly important for scholars, research administrators, and science policymakers. Discovering seminal work creator from scholarly data is crucial to facilitate the understanding of trends and history of a target field. This is especially the case in the field of Computer Science, where more subdomains are constantly born at an extremely rapid growth rate in recent decades. Consequently, it makes it meaningful for researchers to know the source of that specific field. Similar phenomenon also holds in a large number of other scholarly fields such as physics, biology, chemistry and etc. Under such circumstance, it is desirable to have a mechanism that can effectively help researchers find, in those fields of their interests, the seminal work creator.

Literally, finding work creator in scholarly networks is closely related to the problem of source locating in social networks as well as tracking dissemination of relevant information through a social network. While those lines of work have been intensively studied in traditional social networks, there has been little attention given to the seek of the work creator in scholarly fields. Among those that indeed have tie to source locating, existing works target this problem based on heuristic topological centrality measures, maximum likelihood (ML) estimator as well as maximum a posteriori (MAP) framework. For example, with the a priori knowledge of suspect nodes and a snapshot observation of infected nodes, Dong et al. [1] construct a maximum a posteriori (MAP) estimator to identify the rumor source using the susceptible-infected (SI) model. Shah [2] model rumor spreading in a network with the popular susceptible-infected (SI) model and then establish a maximum likelihood (ML) estimator for the rumor source.

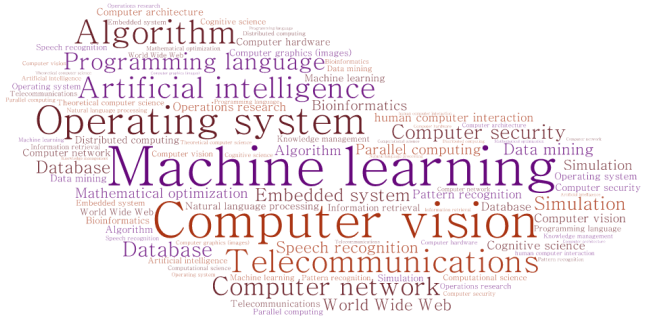

Fig. 1. Words cloud consists of 30 fields in the words cloud, where the place and color of words were random. The fields containing more papers appear with greater prominence.

Despite the delightful predictability of such predictions in conventional social networks, it encounters compromised effectiveness when applied directly to scholarly networks due to the intrinsic difference between social and scholarly networks, primarily for the following reasons. First, in the social networks, most of the source estimators are based on the priori knowledge of suspect nodes. However, there is no reliable datasets of infected nodes for our training. Second, compared to the network of dissemination for rumors in social networks, the networks of paper reference are more centralized. The underlying solution to the these problems requires a largescale scholarly datasets that can provide complete scholarly information, especially the topic information of paper. In light of those difficulties and limitations, in the present work we use a real scholarly dataset from Acemap with more than 1.5 million papers in the CS field, which also contains the hierarchy structure of topics and citation relationship between papers. By investigating all the topics of CS field, our goal is to find out the seminal work creator in the fields.

The main contributions of this work are shown as follows:

- We divide the fields in the computer science with two methods: topic information and text clustering.
- We build a classifier to detect the field of a new paper by deep neural networks.
- We define the influence of paper in a field and find the source seminal work creator in every field.

The remainder of the paper is structured as follows. Section

2 reviews existing work about source locating. Section 3 presents the methods and key factors we use to detect fields. Field detection for new paper is demonstrate in Section 4. Section 5 shows the process of getting the results. Analysis of results are showing in section 6.

II. RELATED WORK

Traditionally, the source locating problem always focusses on capturing the temporal traits of information during their propagation. Kwon et al. [3] introduced a time-series-fitting model based on the temporal properties of a single feature C tweet volume. Ma, et al. [4] extended the model using dynamic time series to capture the variation of a set of social context features over time. Friggeri, et al. [5] characterized the structure of misinformation cascades on Facebook by analyzing comments with links to rumor debunking websites.

There are also some prior studies attempting to classify the veracity of spreading using the text content. Ma et al. [6] using a deep learning model for rumor detection on microblogs. Zang et al. [7] propose a topic-aware source locating method based on topic analysis of propagation items and participants. Our methods firstly applying the deep learning model in the scholarly networks focused on source locating problems based on the text content and fields.

III. FIELD DIVISION

A. Fields division based on topic information in Acemap dataset

Since we build our work based on the Acemap dataset, which provides topic information of each paper and the topic hierarchy structure, we first divide the fields according to these topic information. The hierarchy of topics contains 4 levels in the dataset, L0, L1, L2 and L3. L0 level represents the basic domain of the whole academia, such as Computer Science, Mathematics, Biology etc. We choose Computer Science as our research object. The L0 topic contains L1 topics. To the Computer Science, L1 topics are some basic fields of the CS area, such as Network, Data mining etc. The L2 and L3 topics are also parents and children relationship. Our goal is to find out the L1 field of papers based on their topics.

For the topics in the whole field are not isolated, the connection between topics is not totally one to one, which means children field may belong to several different parents field with confidence. In the dataset, papers have many different topics in different level based on them keywords. Therefore, we count the weight of those topics in L1 level, and choose the top field as the results. The L1 field in computer science are showned in the Fig. 1. There are 29 fields in the words cloud, where the place and color of words were random and the size of words were based on the number of the paper in the field.

B. Fields division based on text clustering

Text clustering is a suitable technique used to partition a huge set of text documents into a predetermined number of clusters. Therefore, based on the texts extracted from Acemap dataset, we also try to detect the fields of paper based on

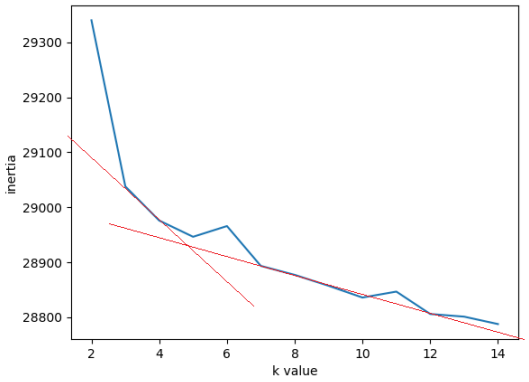

Fig. 2. The performance of kmeans with the different k value. According to the slope of the curve, we find the elbow when k=5.

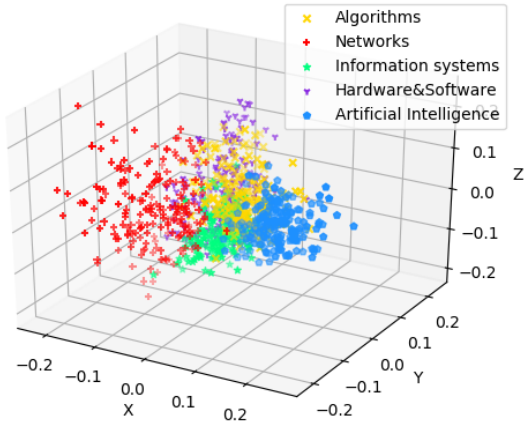

Fig. 3. With the help of PCA methods, we reduce dimensions of vector space, and show the distribution of clusters.

text clustering. Before we apply text clustering technique, it is necessary to convert document contents to become manageable in the algorithm. In this work, we choose the term frequency-inverse document frequency (TFIDF)to convert the document contents in numerical form.

1) term frequency-inverse document frequency: TFIDF is the common weight scheme used to calculate the term weighting in the area of text mining for the document representation. Each document is represented as a vector of terms weights as follow.

$$b_i = (w_{i,1}, w_{i,2}, w_{i,3}, ...)$$

The term weighting is assigned for each term according to the term frequency in each document and others factors. If the term frequency is high and the same term appears in a few documents, we conclude that this term is useful to distinguish among the documents. The term weighting is calculated:

$$w_{i,i} = tf(i,j) * idf(i,j)$$

where $w_{i,i}$ represents the weight of term j in document i, and tf(i, j) represents frequencies of term j in a document i.

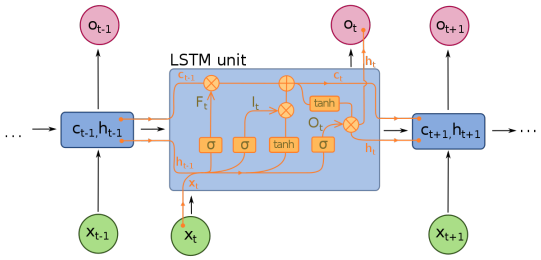

Fig. 4. LSTM

idf(i, j) is a factor used to improve the term which has low frequency and appears in a few documents.

2) Unsupervised text clustering problem: After converting document contents to become vector space as mentioned above, we build the model by k-means as the clustering method. This method is the common clustering method in educational data mining. This is because k-means is the hard clustering. This means that instance or object will be mapped obviously. In this step, we accomplish initializations to get the better result of the clustering process by k-means++, since the clusters quality of this method depends on the initial centroids.

Since the value of k is crucial to the results, we use elbow Method to determine the number of clusters. We run the algorithm for different values of K and plot the K values against the performance in Fig. 2, in which the clustering performance are evaluated by the sum of the central mean vectors of the cluster. Finally, we select the value of K as 5 for the elbow point.

Finally, We define the fields of the clusters by extracting the features in each cluster. With the help of the TFIDF methods, we determined the five fields as follow: Hardware&Software, Artificial Intelligence, Network, Algorithm, Information management. In order to show the results of text clustering, we extract 200 papers in each cluster randomly and calculate the vectors. Since the dataset has a large number of variables, we visualise high-dimensional vector space with PCA method, which tries to provide a minimum number of variables that keeps the maximum amount of variation or information about how the original data is distributed. The distribution of clusters are shown in Fig. 3.

IV. FIELD DETECTION FOR NEW PAPERS

For the whole dataset, which includes more than one million papers, we use classifier to detect their fields. What's more, when a new paper is given, we can also get its field and find its source immediately.

Here we use two kinds of classifiers to achieve this, one is CNN along with word2vec preprocessing the data, another is RNN along with TensorFlow for preprocessing.

A. CNN+word2vec

1) CNN: A convolutional neural network (CNN) is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery. CNN consists of one or more convolution layers and the fully connected layer at the top (corresponding to the classical neural network), and also includes the correlation weight and pooling layer.Here we use an embedding layer, followed by a convolutional, max-pooling and softmax layer. In addition, we use the Softmax function as the loss function.In mathematics, the softmax function is a generalization of the logistic function that "squashes" a K-dimensional vector z of arbitrary real values to a K-dimensional vector $\sigma(z)$of real values, where each entry is in the range (0, 1), and all the entries adds up to 1. The function is given by

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

2) Word2vec: This tool provides an efficient implementation of the continuous bag-of-words and skip-gram architectures for computing vector representations of words. These representations can be subsequently used in many natural language processing applications and for further research.

The word2vec tool 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. The resulting word vector file can be used as features in many natural language processing and machine learning applications.A simple way to investigate the learned representations is to find the closest words for a user-specified word. The distance tool serves that purpose.

There are two main learning algorithms in word2vec : continuous bag-of-words and continuous skip-gram. The switch -cbow allows the user to pick one of these learning algorithms. Both algorithms learn the representation of a word that is useful for prediction of other words in the sentence. Here we use the second one.

3) Result: After we have trained the data we have clustered, we classified the whole dataset. The result is shown in Fig. 5

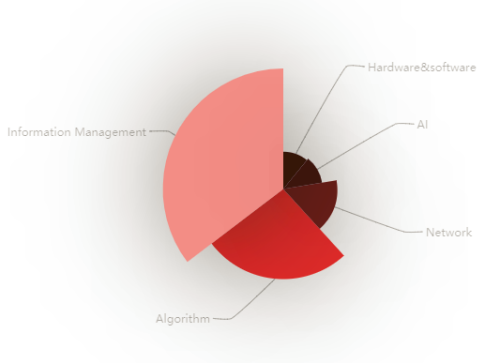

Fig. 5. Paper distribution with CNN

B. RNN+TensorFlow

1) RNN: A recurrent neural network is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike feed-forward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. And to calculate the hidden state, we use LSTM.LSTM is a deep learning system that avoids the vanishing gradient problem. LSTM is normally augmented by recurrent gates called "forget" gates. LSTM prevents backpropagated errors from vanishing or exploding. Its structure is shown below.

2) TensorFlow: Tensor, in my understanding, is just the matrix. It can also be understood as the matrix representation in tensorflow. And for the data preproccessing, we achieve the same effect as word2vec through Tensorflow actually.

- First, we split the data into words and generates labels and pad all sentences to the same length.
- Then, we build a vocabulary mapping from word to index based on the sentences.
- Last, we map sentences and labels to vectors based on the vocabulary.

With the process above, we can achieve the same effect as word2vec more quickly and conveniently.

3) Result: After we have trained the data we have clustered, we classified the whole dataset. The result is shown in Fig. 6.

V. RESULTS

As the fields of the paper have already prepared, we are trying to find the source in each field. There are two features to estimate the possibility of a paper to be the seminal work creator, respectively, published year and influence. Therefore, we will prepare the top 5% papers based on the influence as candidates and select the paper that published earlist.

In this work, we are trying to judge a paper influential based on the following features: a paper is cited by many papers, or it is cited by influential papers. Apparently, papers with a lot of citations are more influential than those with few citations. With regard to papers with same citation counts, the papers cited by influential ones are more likely to be influential than those cited by regular papers. Therefore, we use the PageRank

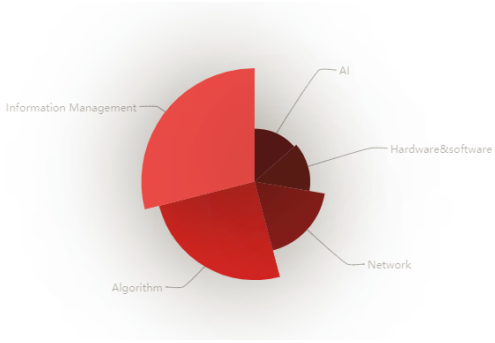

Fig. 6. Paper distribution with RNN

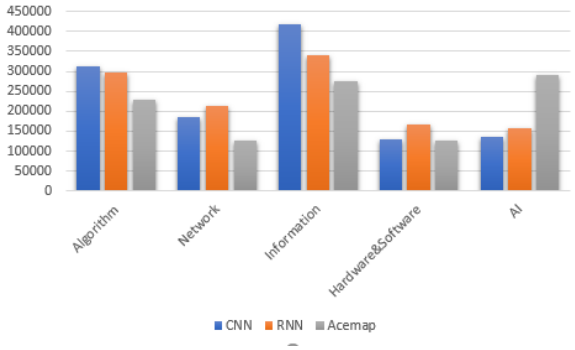

Fig. 7. Counting up numbers of each field for three method

method that are proposed by Xie et al. [8] to weighting paper influence objectively.

The PageRank method is originally used for link analysis by search engines, and it proposes that a web page itself carries a greater importance if linked to by other high importance pages. The PageRank of a paper is defined depends on the number and PageRank metric of all papers that cite to it. A paper that is cited by many papers with high PageRank receives a high rank itself. We adopt the PageRank algorithm to calculate the influential of academic papers as follow:

$$PR(p_i) = d + \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

where$p_j \in P$ is an academic paper, d is the dampening factor, $M(p_i)$ is the set of all inbound citations to pi, $L(p_j$ is the number of outbound citations of pj.

The seminal work creator we have detected with three methods in every field, the title and published year are also shown in TABLE I.

VI. ANALYSIS

A. Analysis of Field

In this work, we divide the fields in the computer science with two methods: topic information and text clustering. The topic information we obtained from Acemap datasets mainly based mainly on the paper keywords, making the fields are more reliable. However, there are 29 L1 fields in the computer science and some of them are very similar, such as Computer graphics (images) and Computer vision. The similarity of fields will influence the performance of classifier and source detection. Moreover, since we define the fields of paper with the weight of those topics and one paper may have dozens of topics distributing in different fields, it is possible that paper may be categorized into wrong fields. Given the factors I have just outlined, the results in the fields of Telecommunications and World Wide Web have not done very well.

We also use the method of text clustering. The fields are less but more clear, leading the better results of source detection. However, in an academic paper, commonly used words with ambiguous meanings are more likely to be selected due to their frequent appearance while the technical terms with

TABLE I

THE SEMINAL WORK CREATOR WE HAVE DETECTED WITH THREE METHODS IN EVERY FIELD, THE TITLE AND PUBLISHED YEAR ARE SHOWN AS FOLLOW.

Field	Title	Published Year	PageRank value
cnn methods			
Network	Shortest connection networks and some generalizations	1957	2.27
Information management	A General Switching Plan for Telephone Toll Service	1930	1.13
Artificial Intelligence	Pattern Detection and Recognition	1959	1.46
Algorithm	Theory of Frequency-Modulation Noise	1948	0.98
Hardware&Software	High-Speed Arithmetic in Binary Computers	1961	1.97
rnn methods			
Network	Shortest connection networks and some generalizations	1957	2.27
Information management	Communication theory of secrecy systems	1949	2.95
Artificial Intelligence	Pattern Detection and Recognition	1959	1.46
Algorithm	Fast Carry Logic for Digital Computers	1955	7.20
Hardware&Software	High-Speed Arithmetic in Binary Computers	1961	1.97
Acemap dataset			
Computer hardware	An approach to the implementation of digital filters	1968	3.93
Operating system	A study of replacement algorithms for a virtual-storage computer	1966	1.78
Operations research	Scheduling of Vehicles from a Central Depot to a Number of Delivery Points	1964	1.29
Computer graphics (images)	A fast procedure for computing the distance between complex objects in three-dimensional space	1988	2.63
Artificial intelligence	AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly	1977	1.14
Real-time computing	Speech enhancement using a soft-decision noise suppression filter	1980	4.70
Pattern recognition	Programming pattern recognition	1955	5.05
Computer vision	The internal representation of solid shape with respect to vision	1979	1.14
Simulation	A statistical discrete-time model for the WSSUS multipath channel	1992	2.98
Machine learning	The automatic creation of literature abstracts	1958	1.18
Computer architecture	An architecture for reflexive autonomous vehicle control	1986	1.81
Database	Query-by-Example: A data base language	1977	0.97
Computer network	Shortest connection networks and some generalizations	1957	2.27
Natural language processing	Three models for the description of language	1956	1.12
Embedded system	Fast Carry Logic for Digital Computers	1955	7.20
Computational science	Computational proxies: modeling scientific applications in object databases	1994	1.29
World Wide Web	A human machine interface for distributed virtual laboratories	1994	1.97
Algorithm	ECG data compression techniques-a unified approach	1990	1.47
Parallel computing	A study of non-blocking switching networks	1953	2.82
Information retrieval	A Statistical Approach to Mechanized Encoding and Searching of Literary Information	1957	1.96
Telecommunications	Capacity of a burst-noise channel	1960	6.49
Cognitive science	A system for aiding creative concept formation	1994	1.88
Knowledge management	Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation	1991	4.16
Data mining	Epoch extraction from linear prediction residual for identification of closed glottis interval	1979	0.95
Speech recognition	An algorithm for automatic formant extraction using linear prediction spectra	1974	2.01
Mathematical optimization	The Truck Dispatching Problem	1959	2.03
Programming language	The LISP 2 programming language and system	1966	1.39
Human computer interaction	Use of Model-Based Qualitative Icons and Adaptive Windows in Workstations for Supervisory Control Systems	1987	1.76
Computer security	Communication theory of secrecy systems	1949	2.95

significantly lower occurrence frequency can be easily missed out. Therefore, we are not able to divided the small-scale field, such as bioinfomatics and signal processing.

B. Analysis of Classifier

For we define the field by ourselves, there's no suitable test data. Based on the word of the most weight in every paper(get the weight from tfidf), we compared the fields in our project and Acemap, and get the corresponding relations between the two. After counting up the number of each field, we compare the two with Acemap dataset, as in Fig. 7.

We can see from the figure that most of them are similar, while AI field are extremely different. We guess that AI field includes so many sub-field that many papers in AI field actually tell about other things, but for AI nowadays is indeed hot, many fields contain its techniques. From these we can explain why the difference come out.

VII. CONCLUSION

Most existing work on source locating focus on social media through extracting features or rules manually. In this work, we propose a model to tracking the seminal work creator in big scholarly networks. We detect the field with two methods and provided the analysis of the results. There is still some future work worth studying. While we conducted our work in the CS field, it is necessary to examine and observe the results in other science fields such as mathematics, biology, literature and so on. About the source detection, the correlations between published year and influence have more valuable informations, to know how they affect each other, more work is needed.

Name	Student Num.	Team Work
Mengxin Liu	515021910215	data extraction and preprocess, Field detection(section 3), Source Locating(section 5)
Wencheng Tang	515021910424	Preprocess and classify the whole dataset with two methods (CNN+word2vec and RNN+Tensorflow), which enable a new paper to find its field and source(section 4).

TABLE II

REFERENCES

[1] Dong W X, Zhang W, Tan C W. Rooting out the rumor culprit from suspects[J]. 2013:2671-2675.

[2] Shah D, Zaman T. Rumors in a Network: Who's the Culprit?[J]. IEEE Transactions on Information Theory, 2009, 57(8):5163-5181.

[3] Kwon S, Cha M, Jung K, et al. Prominent Features of Rumor Propagation in Online Social Media[C]// IEEE, International Conference on Data Mining. IEEE, 2014:1103-1108.

[4] Ma J, Gao W, Wei Z, et al. Detect rumors using time series of social context information on microblogging websites[C]//Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 2015: 1751-1754.

[5] Friggeri A, Adamic L A, Eckles D, et al. Rumor Cascades[C]//ICWSM. 2014.

[6] Ma J, Gao W, Mitra P, et al. Detecting Rumors from Microblogs with Recurrent Neural Networks[C]//IJCAI. 2016: 3818-3824.

[7] Zang W, Zhou C, Guo L, et al. Topic-aware source locating in social networks[C]//Proceedings of the 24th International Conference on World Wide Web. ACM, 2015: 141-142.

[8] Xie, Yi, Yuqing Sun, and Lei Shen. "Predicating paper influence in academic network." Computer Supported Cooperative Work in Design (CSCWD), 2016 IEEE 20th International Conference on. IEEE, 2016.