Building the Extraction Model of the Software Entities from Full-Text of Research Articles Based on BERT

Chuan Jiang[1[0000-0003-2436-9411]] Dongbo Wang[1[0000-0002-9894-9550]] Si Shen[2[0000-0002-6990-410X]]
Wenhao Ye[3[0000-0003-2811-4248]] Jiangfeng Liu[1[0000-0001-7268-7313]]

[1] College of Information Management, Nanjing Agricultural University, Nanjing 210095, China
[2] School of Economics & Management, Nanjing University of Science & Technology, Nanjing 210094, China
[3] School of Information Management, Nanjing University, Nanjing 210023, China

Abstract. Software entities in the full-text of research articles are vital academic resources. Extracting software entities from the full-text of research articles can improve the process of knowledge organisation and is a important aspect of knowledge entitymetrics. In this study, the full-text of research articles were collated from the journal *Scientometrics* from 2010 to 2020. The extracted software entities are subjected to metric analysis and mining from different perspectives, such as the distribution in the different structures of articles, the number of mentions and citations, and the time-series evolution. To build an automated software entities extraction model, entitymetrics tools are provided. The machine learning and deep learning models, namely, conditional random field (CRF), Bi-LSTM-CRF, and the bi-directional encoder representation from transformers (BERT), were established. Tthe highest F1 values of 84.99% was achieved with BERT. The future implications of the study include the application of the BERT-based model for the software entity extraction from other journals to deepen the mining and the analysis of software entities from multiple perspectives.

Keywords: Deep learning; BERT; Full-Text; Entitymetrics

1 Introduction

The development of the full-text of research articles data has been complemented by the continuous progress in the data analysis and extraction technology. The demand for the deep mining of the full-text of research articles has increased in the field of bibliometrics and knowledge organisation owing to the improvement in the metric analysis and the visualisation research of the literature metadata. Consequently, the academic research of the full-text of research articles has gained widespread attention.

Software is a crucial aspect of academic research, as it is essential to facilitate academic findings and interdisciplinary exchanges and collaboration, due to which most of the current research is driven by data and software (Nangia and Katz 2017). Ex-

tracting the software entities from the full-text of research articles innovative the form of knowledge organisation and the key of knowledge entitymetrics for the full-text of research articles (Ding et al., 2013). Currently, the software entities are predominantly extracted from metadata and footnotes, and they are generally extracted by using dictionaries and heuristic rules. However, these extraction methods typically encounter drawbacks, such as low data volume and limited extraction performance. Therefore, the extractions methods with greater accuracy are required for bibliometrics.

To fill the gap, a novel corpus technology has been presented in this study to develop the corpus for software entities extracted from the full text of the research articles from *Scientometrics* from 2010 to 2020. The software entities were extracted from *Scientometrics* using the BERT model and the latest deep learning model for natural language processing (NLP). These entities were subjected to metric analysis and mining from different perspectives, such as the distribution in the different structures of articles, number of mentions and citations, and time-series evolution. The distribution laws of the software entities extracted from *Scientometrics* were discussed. Consequently, the research trend and the objects in Scientometrics were retrospectively analysed from 2010 to 2020.

2 Literature Review

2.1 Content Analysis of Full-Text of Research Articles and Its Applications

The content analysis of the full-text of research articles involves deep mining of the structure of the articles, citations, lexical style, syntax, and the topics. It can be combined with conventional bibliometrics to improve the reliability and the accuracy of the latter. The content analysis of the full-text of research articles is based on the citation, linguistic, and thematic perspectives.

Citation Perspective. Combining the conventional citation count with content analysis can increase both the depth and the width of the citation content analysis and improve the clustering performance. Zhang et al. (2013) proposed a new framework for the citation content analysis, which employs NLP techniques to analyse the syntax and the semantics of the documents. This framework was suitable to analyse the characteristic citation style and the background behind the citation. Jeong et al. (2014) collected the full-text of the research articles from JASIST and proposed an improved method for the author co-citation analysis. This method involved measuring the similarities between the authors by considering the citation contents and presented more details than the conventional method. Hu et al. (2017) collected the full-text of the research articles data from the *Journal of Informetrics* to analyse the condition where the citations were mentioned repeatedly in the full-text of research articles, such as lengthy research articles and self-citations. Moreover, these citations were usually repeated in the same section of a research article. Small et al. (2017) acquired the full-text research articles data from PubMed and analysed the contextual sentences for the citations through lexical analysis and machine learning. This method was intended to

be applied to the knowledge discovery process in biomedicine. To study the distinctions and the relationship between the highly cited methodology and the non-methodology papers, Small (2018) collected the full-text of the research articles from PubMed. Methodology and non-methodology papers in biomedicine were differentiated with the help of the corpus technology and with machine learning. It was observed that the methodology papers were predominant among the highly cited papers. Bu et al. (2018) collected the full-text of the research articles from JASIST. They improved the conventional author co-citation analysis by introducing the number of citation mentions and the contextual vocabulary for citations. The improved method presented a better clustering performance.

Linguistic Perspective. The full-text of research articles can be analysed by the NLP techniques and by the linguistic methodology to understand the style, language habits, and text structure. Yan and Zhu (2018) collected article abstracts from PubMed to explore the semantic variations of the words used in biomedical literature and applied the Word2Vec model and the topic model to this end. Li et al. (2018) collected the abstract data from WOS and extracted the verbs and the nouns from the WOS-related sentences. They observed that WOS was usually mentioned as a database. To study the language distribution in non-formal academic exchanges, Yu et al. (2018) collected the full-text of the research articles from Scopus and Tweets. They reported an interdisciplinary difference in the language distribution in academic tweets and literature and concluded that English had become a common language in non-formal academic exchanges. To reveal the difference between non-native and native English-speaking scholars in writing, Lu et al. (2019) collected the full-text of the research articles from a *PloS* journal. They attempted to differentiate the writing style based on the syntax and the lexical perspectives. Thelwall (2019) collected the full-text of the research articles from PubMed Central. They reported substantial differences in the article structure between the literature of various fields.

Topic Analysis Perspective. Topic mining of the full-text of research articles can derive more information than the conventional methodology and exhibits better performance. Glänzel and Thijs (2017) applied the NLP techniques and clustering to the full text of the research articles in astronomy and astrophysics to characterise and recognise the corresponding topics and clusters. To identify the preferred topics in the bibliometrics journals, Zhang et al. (2018) collected the full-text of the research articles from the top three journals in bibliometrics and proposed a topic extraction method that combined K-means and Word2vec. Thijs and Glänzel (2018) acquired the full-text of the research articles from *Scientometrics* and applied the Stanford NLP method to extract noun phrases from the full-text. They studied the contribution of the vocabulary composition to the hybrid clustering of the topics in the full-text of research articles. They reported consistency between the word clusters and the bibliographic coupling.

The content analysis of the full-text of research articles integrated with conventional bibliometrics is beneficial in establishing prominent findings in research articles and improve the performance of conventional extraction methods.

2.2 Entitymetrics and Knowledge Discovery Based on Full-Text of Research Articles

In recent years, there has been a tremendous increase in the volume of full-text research articles data along with the progress in analysis and extraction techniques. Increasing attention has been drawn to fine-grained entitymetrics and knowledge mining of the full-text of research articles.

Entitymetrics. The concept of entitymetrics was first proposed by Ding et al. (2013) as a measure of the influence of the knowledge unit. Entitymetrics highlights the importance of entities in the process of knowledge discovery. The entities are further divided into knowledge entities and evaluation entities. The recent studies conducted on entitymetrics have been primarily concerned with the metrics of the knowledge entities, including software, data resources, algorithms, models, domain entities, and terms. Dictionaries and heuristic rules are the commonly used methods to extract entities from the full-text of research articles. A majority of the academic researchers currently use software for research. Some scholars have found through surveys that software entities often have a low citation rate and software is shared only in a few specific disciplines (Park and Wolfram 2019). Li et al. (2017) investigated the citation of the R package to account for the granularity of the software entities and observed that the citation of the R package was inconsistent with the R software. The authors proposed the fine-grained citation of the single functions of the software environment and the package. Recent studies based on the software entity metrics have considered the full-text of the research articles from *PlosOne* as the object in general. The software entities are extracted by using various methods, such as bootstrapping, dictionaries, and machine learning. The influence of software entities is assessed in terms of citations, mentions, and frequency, and the relationship between the packages is analysed by developing an entity network (Yan and Zhu 2015; Yan and Pan 2015; Pan et al. 2015; Zhao et al. 2018). Wang and Zhang (2018) extracted algorithmic entities using dictionaries and rules from the full-text of the research articles from ACL. The academic influence of the algorithmic entities in the field of NLP was then studied by considering the number and the position of the mentions. They reported that the support vector machine (SVM) exhibited greater influence. Zhao et al. (2018) collected the full text of the research articles from *PlosOne* to analyse the mentions and citations of the database entities, and they observed that the dataset reuse rate was less than 30%. This indicates that the researchers tend to develop their own datasets. Ding et al. (2013) extracted the entities such as genes and drugs from the full-text of research articles in PubMed to account for the field-specific terms and entities. They developed the field-specific entity citation network to assess the influence of entities in the biomedical field. Chen and Luo (2019) collected textual data from the Web of

Science and Scopus databases, and entities were extracted from the abstracts by using the BERT model. A network for inference on the scholarly knowledge graph was developed to enhance the metric analysis of the research articles. The above studies are primarily focused on specific fields. Among the studies on the extraction of field-irrelevant terms, Yan et al. (2017) applied the NLP techniques and rules to extract the full-text of the research articles from PlosOne. They extracted the academic terms of various disciplines and discovered a power-law pattern in the frequency distribution of the terms in each discipline. Chen and Yan (2017) employed the NLP techniques and scoring rules to extract terms from the abstracts of research articles. A term network was developed to analyse the importance of multi-field terms and their time-series evolution.

Knowledge discovery. Entity extraction from the full-text of research articles can facilitate the assessment of the influence of entities and also contribute to knowledge discovery. Current studies on entity extraction for knowledge discovery are mainly limited to the biomedical field wherein the use of dictionaries and machine learning is preferred. The correlations between the knowledge entities are mined by network analysis. Lv et al. (2018) collated the abstracts of papers on autism from PubMed to analyse the topological correlation between the drugs for autism and to mine the recent trend in the drug development. The drug entities were extracted based on the MeSH Translation Table, and the drug entity network was then developed. Yu et al. (2015) applied dictionaries to extract the medical database entities from PubMed. They developed a database link network to analyse the topological structure and the main paths of the network based on which the database use, link, and evolution were tracked. Song et al. (2015) extracted the biomedical entity correlations from the abstract dataset of PubMed by using machine learning and rules. They developed a biomedical entity network and proposed a semantic path-based method for biomedical knowledge discovery. Song et al. (2013) used the conditional random field (CRF) and the Unified Medical Language System to extract the gene entities from the abstracts of papers in MEDLINE. The gene entity network was established based on the citation correlations. The interactions between the gene entities were identified by network analysis.

Studies conducted on entitymetrics based on the full-text of research articles generally consider knowledge entities as objects (e.g., software). Meanwhile, studies conducted on knowledge discovery are primarily focused on the biomedical field, where the entities are extracted by rules and dictionaries. Although such extraction methods exhibit high precision and speed, their recall is low, and they generally fail to recognise new entities in the field. They also face the drawbacks of small data size and limited number of open datasets; that is, the data are insufficient for large-scale, accurate entitymetrics and knowledge discovery.

3 Metric Analysis of Software Entities

The data used in this study were obtained from the papers of *Scientometrics* from 2010 to June 2020. A Web crawler was employed, and 3,522 papers published in *Scientometrics* were collected in total, as shown in Fig. 1.

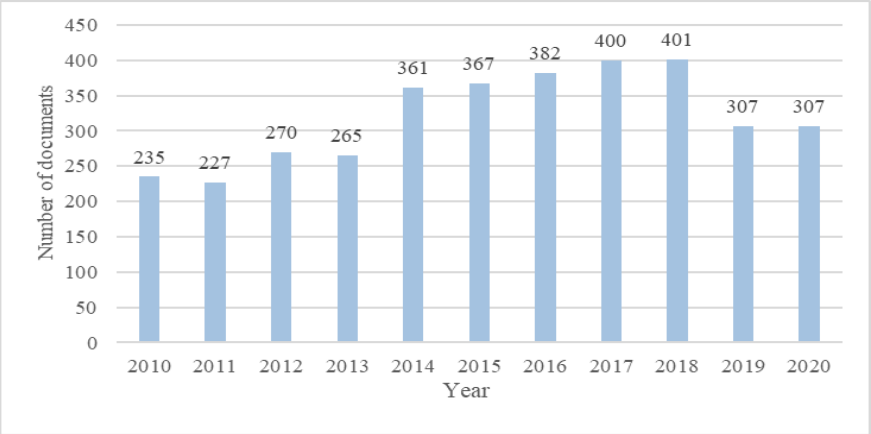

Fig. 1. Distribution of scholar documents in *Scientometrics* from 2010 to 2020

The PyQuery package (PASGRIMAUD 2017) was used for the parsing of the HTML full text of research articles. The letters, reports, and inaccurate and blank data were eliminated. Thus 3,493 full-text of the research articles including 156,318 paragraphs were obtained and stored in the Mysql database.

For the full text data that has been acquired, this research uses the Brat annotation platform (Stenetorp et al. 2012) to label the software entities of academic full texts. As shown in Fig. 2, the Brat is currently a widely used the field of natural language processing, which can be used to label entities, entities relationships and structure syntax tree, etc. This research first uses the StanfordCoreNLP toolkit (Manning et al. 2014) to segment the paragraphs in the MySql database, converts the collection of academic full-text chapters into a collection of sentences, and then imports it into the Brat annotation platform to construct the *scientometrics* software entity corpus by manual annotation. Based on this, the software entity extraction model is constructed, and the wrong software entities are corrected by analyzing the difference between the prediction results of the software entity extraction model and the corpus labeling results. Finally, we obtained the scientometrics software entity annotation corpus.

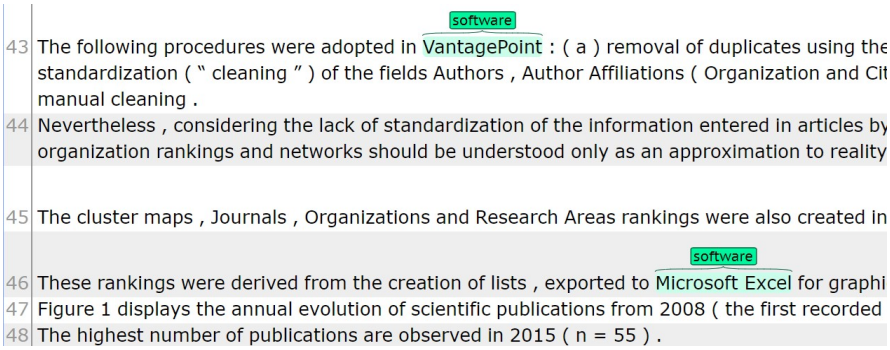

Fig. 2. The example of manual annotation software entities in Brat

In order to better understand the software entities in the academic full text, we provide 3 sentences containing software entities, where the software entities in the sentences are marked with <software>, such as:

- The patent networks were drawn by using <software>UCINET 6.0</software> (Borgatti et al.). In the following section, we describe the structural features of patent networks in overall network and cluster levels.
- After extracting binary relations that appear together in each sentence of patents using the <software>Stanford parser</software>, unintended or too-general binary relations are filtered out using English stopwords (STOPWORDS).
- Full-text search is supported using <software>Solr</software> (The Apache Software Foundation) to index the contents of the database.

3.1 Distribution of Software Entities in Different Structures of Articles

The distribution of the software entity mentions in different structures of the articles were further analysed in this study. The collected full-text of the research articles were divided into several parts according to the common classification method for the structure of the articles proposed by Ding et al. (2013). The collected documents were divided into the following parts by the manual annotation process: Introduction, Related Work, Method, Experiment and Result, and Discussion and Conclusion. As shown in Fig. 3, the software entities were predominantly mentioned in Method and Experiment and Result. This distribution pattern corresponds with the general organisation of a research article. In research papers, the tools and the software are generally introduced in these two sections, with the steps of the software implementation occasionally being explained in greater detail. The software entity mentions also appear frequently in Related Work, where the previous studies involving the use of the relevant software are generally reviewed. Conversely, the Introduction, and Discussion and Conclusion primarily describe the significance and the contribution of the study and do not often mention the software entities.

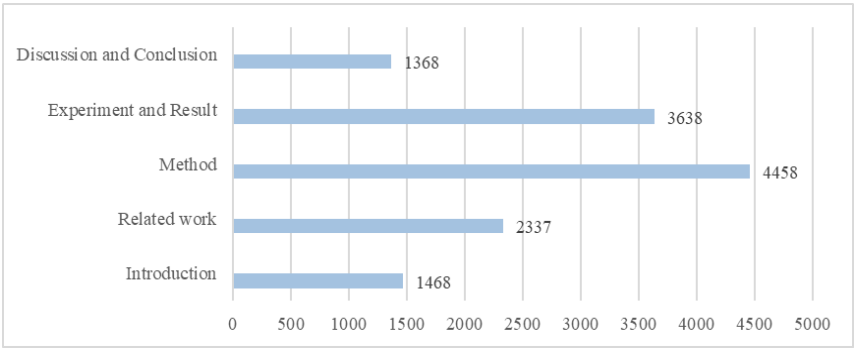

Fig. 3. Distribution of the number of software entity mentions in different structures of articles

3.2 Mentions and Citations of Software Entities

The number of software entity mentions in the full-text of the research articles from *Scientometrics* from 2010 to 2020 was analysed to determine whether recent academic efforts were driven by software. According to Fig. 4, the number of software entity mentions had been increasing over the years, with a peak of 2,047 mentions in 2018. This result indicates that *Scientometrics* has attached greater significance to the use of software when compared to power academic studies.

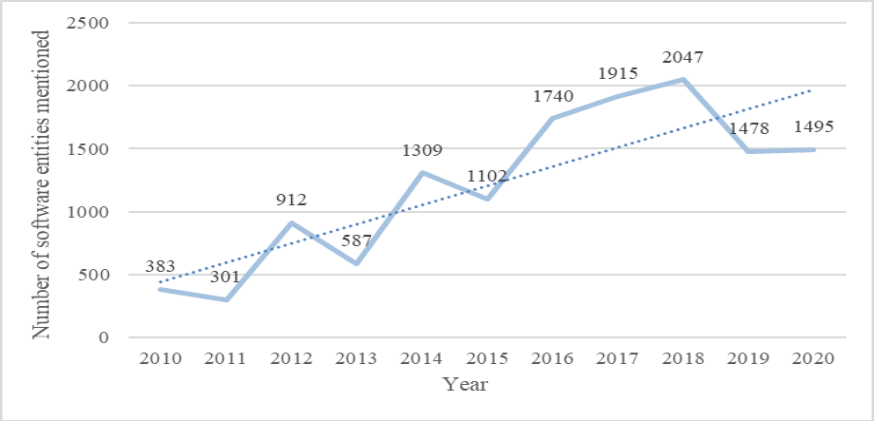

Fig. 4. Distribution of the number of software entity mentions in *Scientometrics*

The number of software entity mentions has been increasing in the recent years, with software entities becoming vital academic knowledge in research articles. The standardised citation of software entities is intended to recognise the software developers and is an academic norm that must be conformed to. The distribution of the number of software entity citations in *Scientometrics* was analysed from 2010 to 2020. As shown in Fig. 5, the number of software entity citations has comprehensively increased over the years. This indicates the increasing significance attached to the standardised software citations in the research articles.

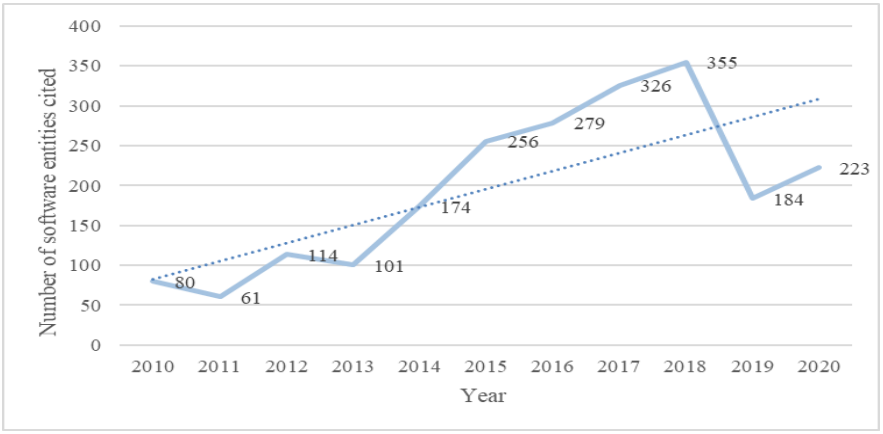

Fig. 5. Distribution of the number of software entity citations in *Scientometrics*

The correlation between the number of software entity mentions and the corresponding number of documents was analysed from 2010 to 2020, as shown in Fig. 6. A power law was observed in this correlation over the years. In particular, about 80% of the research articles had seven software entity mentions and below; only 20% of the research articles had over seven software entity mentions. This finding was consistent with the conclusion from the data research of *PLoSONE* by Pan et al. (2015).

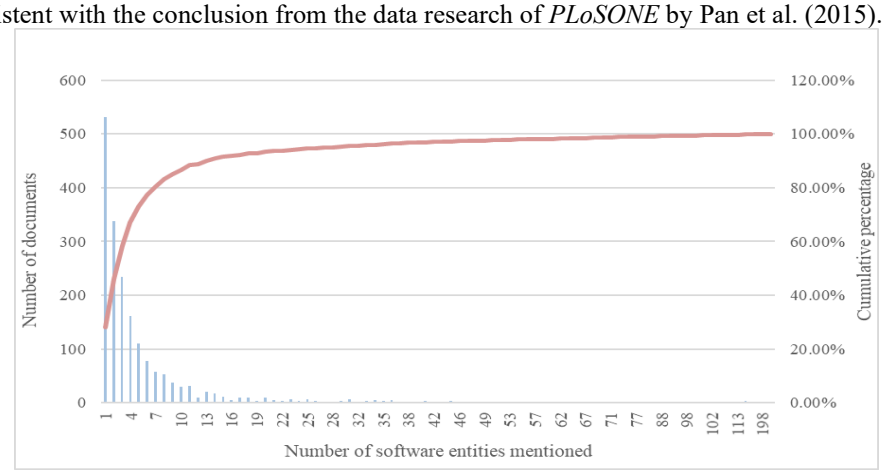

Fig. 6. Relationship between the number of software mentions and the number of documents

3.3 Time-Series Analysis of Software Entities

The top 10 software entities in terms of the number of mentions were analysed from 2010 to 2020. The time-series analysis presented the research trends and the objects in the papers published in Scientometrics within the past decade. Consequently, a basic concept of the mainstream methodology used in academic research can be obtained and the prevailing topics in this field can be reviewed. Manual proofreading was per-

formed for the extracted software entities to obtain statistics with greater accuracy. The problems such as inconsistency in the version number, capitalisation, and the software abbreviations and acronyms were suitably rectified, for example, “AMOS, AMOS 22,” “JAVA, Java, java,” and “Excel, MS Excel.”

Table 1. Top 10 most frequently mentioned software entities from 2010 to 2015

No.	2010	2011	2012	2013	2014
1	VOSviewer	CiteSpace	Twitter	SCIgen	Twitter
2	Pajek	UCINET	Google	SPSS	Mendeley
3	Google	SPSS	CiteULike	Excel	Google
4	SPSS	VOSviewer	Mendeley	Twitter	Pajek
5	Excel	Excel	SPSS	FaceBook	UCINET
6	ASE tool	Google	UCINET	UCINET	VOSviewer
7	UCINET	Pajek	Yahoo	Pajek	Excel
8	Network Work Bench	Vantage-Point	VOSviewer	Yahoo	SPSS

As presented in Tables 1 and 2, the Top 10 software entities in terms of the number of software entity mentions from 2010 to 2020 included the bibliometrics tools, VOSviewer and CiteSpace; the software packages for the analysis of website data, Pajek and UCINET; the data analysis tools, Excel, SPSS, and R; and the citation management tools, Mendeley and CiteULike. Google and Twitter generally use APIs to access metadata and full texts of research papers. An API is an interface that facilitates interactions between various software programs. An API has been employed in this study as well.

VOSviewer was mentioned 10 times in 11 years and was not mentioned only in 2013. CiteSpace was mentioned five times. This result indicates that the bibliometrics research papers published in *Scientometrics* preferred the use of VOSviewer. Among the website data analysis tools, Pajek was mentioned eight times in 11 years and UCINET seven times. This indicates that Pajek and UCINET were the mainstream software for analysing website data. SPSS and Excel were the primary tools for journal sorting and statistical analysis used in the research papers from *Scientometrics*. These two software packages were mentioned 10 times in 11 years.

Table 2. Top 10 most frequently mentioned software entities from 2016 to 2020

No.	2016	2017	2018	2019
1	Twitter	Twitter	Mendeley	Mendeley
2	Mendeley	Mendeley	Twitter	Google
3	Google	Google	FaceBook	Twitter
4	VOSviewer	FaceBook	VOSviewer	FaceBook
5	Excel	VOSviewer	Google	VOSviewer
6	FaceBook	Excel	Pajek	ScientoPy
7	Pajek	Python	Excel	Excel
8	Bing	R	CiteULike	Google Trends

The Altmetrics research has undergone continuous development in recent years, as complements to bibliometrics research. Certain citation management tools are used in combination with the data from Twitter and Google. Research papers and the academic influence of the authors can thus be better analysed by incorporating the social media and website analysis. This combination can make up for the drawbacks faced by the conventional citation analysis methods. The reference manager Mendeley was mentioned 10 times, which was five times larger than that of CiteULike. Thus, Mendeley was more favoured in the Altmetrics research. The social media platforms, Twitter, Facebook, and Google were the top 3 in terms of the number of mentions. They represented the latest trend in *Scientometrics*, which appears to favour Altmetrics in recent years.

Many other new software packages appeared from 2010 to 2020, although they were mentioned less frequently in the past 11 years. For example, SCIgen is a program that generates random computer science research papers automatically, including graphs, figures, flow charts, and citations. Certain niche software products, such as the ASE tool, PoP (Publish or Perish Software), and Sci2 (Science of Science Tool), are now available for citation analysis research. Recently, Python has become a mainstream software in the data science analysis. It was one of the top 10 software entities mentioned in *Scientometrics* in 2017 and 2020. ScientoPy has also been developed as an open-source Python-based scientometric analysis tool. From the emergence of Google+, which is Google's social media platform, in 2020, it can be inferred that Google+ may become a new research object for Altmetrics research.

4 Models

The machine learning and deep learning models will be used in the follow-up of this research to build the model of extracting software entities with excellent performance.

4.1 Conditional Random Field

The conditional random field (CRF) (Lafferty et al. 2011) is a popular method used to perform NLP tasks, such as entity recognition, word segmentation, and part-of-speech

tagging. Here, the software entity extraction from the full-text of the research articles in *Scientometrics* was converted into the sequence tagging task. The formal formulae, 1-2, were applied, where X = {X1, X2…. Xn-1, Xn}, represents the character of each sentence in the full-text of the research articles from *Scientometrics*. Consider the example "Two other computer programs that can be used to construct graph-based maps are CiteSpace." Y = {Y1, Y2…Yn-1, Yn} is the tag for the character in each sentence, such as the start, middle, and end tags for software entity characters.

The tag sequence is modelled with X, which is given for the CRF. The formal formulae are expressed in 1–2. The two-valued eigenfunctions, t_k and s_l, are used to extract the character-level features from the full-text of the research articles in Scientometrics. λ_k and μ_l are the weights of the eigenfunctions, which are dynamically adjusted in the software entity extraction model. Z(x) is a normalisation factor that is used to ensure that the conditional probability of P(y|x) falls within the range of 0–1. P(y|x) is the overall score of the entity tags corresponding to the characters in the full text of research articles.

$$P(\mathrm{y}|\mathrm{x}) = \frac{1}{Z(\mathrm{x})}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l}\mu_l s_l(y_i, x, i)\right) \quad (1)$$

$$Z(\mathrm{x}) = \sum_{\mathrm{y}}\exp\left(\sum_{i,k}\lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l}\mu_l s_l(y_i, x, i)\right) \quad (2)$$

4.2 Bidirectional LSTM-CRF

Long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) is a special type of recurrent neural network. The architecture of the bidirectional LSTM, as shown in Fig. 1, can be used to extract contextual information and improve the software entity extraction performance. The neurons of the LSTM were calculated by using formulae 3–8, where i_t, o_t, f_t, and c_t are the control matrices for the input gate, the output gate, the forget gate, and the memory cell of the t-th input character in the full-text of the research articles from *Scientometrics*. These matrices are used to control the input, the output, the forget, and the memory functions of the information in the full-text of the research articles. x_t and h_t are the embedding vectors for the t-th character and the output vector of the hidden neuron at the t-th moment. Further, w and b are the untrained weight vectors, and σ is the Sigmoid activation function.

$$f_t = \sigma(W_{\mathrm{f}} * [h_{t-1}, x_t] + b_{\mathrm{f}}) \quad (3) \qquad \mathrm{i}_{\mathrm{t}} = \sigma(W_{\mathrm{i}} * [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\mathrm{o}_t = \sigma(W_{\mathrm{o}} * [h_{t-1}, x_t] + b_{\mathrm{o}}) \quad (5) \qquad \tilde{c}_{\mathrm{t}} = \tanh(W_c * [h_{t-1}, x_t] + b_c) \quad (6)$$

$$\mathrm{c}_{\mathrm{t}} = \mathrm{f}_{\mathrm{t}} \otimes \mathrm{c}_{\mathrm{t\text{-}1}} + i_t \otimes \tilde{c}_{\mathrm{t}} \quad (7) \qquad h_t = O_t \otimes \tanh(c_t) \quad (8)$$

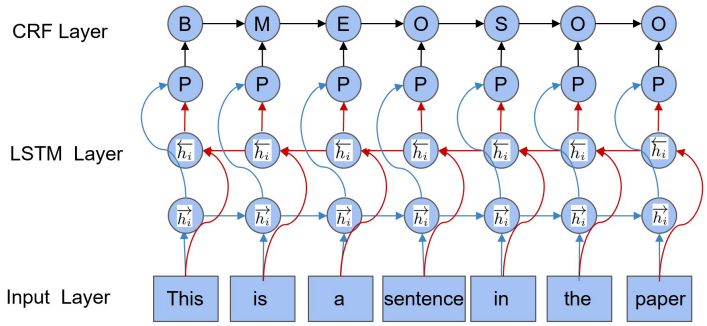

Fig. 7. Architecture of the Bi-LSTM-CRF model for software entity extraction from *Scientometrics*

The bi-directional LSTM was used to model only the features of the full text of the research articles from *Scientometrics* and not those of the software entity labels. Therefore, we introduced the CRF layer (Huang et al. 2015) to model the labels, with the architecture shown in Fig. 2. The CRF layer was able to effectively reduce the errors in the independent prediction of the tags. In particular, the start tag of the software entity is more likely to be the end tag or the middle tag. The formal formula is shown in formula. 9, where $P \in R^{n*k}$ is the entity tag score for the features of the input full-text of the research articles. This score is the probability value of the tag corresponding to the n-th word, where n is the number of words in the current input sentence and k is the tag number; $A \in R^{k*k}$ is the transition probability matrix of the tag. The CRF layer identifies and optimises the status transition probability matrix A and uses Viterbi for decoding. The highest tag score of the entire document is thus obtained along with the global optimal solution, y={y1…yn-1, yn}.

$$S(\mathrm{x}, \mathrm{y}) = \sum_{t=1}^{n+1} (A_{y_{t-1}, y_t} + P_{t, y_t}) \tag{9}$$

4.3 Bidirectional Encoder Representation from Transformers

The bi-directional encoder representation from transformers (BERT) is a deep language representation model based on modifying the common bi-directional language model (Devlin et al. 2018). A transformer encoder based entirely on a self-attention mechanism was used to model the full-text documents from *Scientometrics*. BERT is superior to other neural network models as it is pre-trained by large-scale unsupervised corpus. During the software entity extraction process from the full-text of the research articles in *Scientometrics*, BERT only needs to be fine-tuned with the top-level parameters to predict the software entity tags.

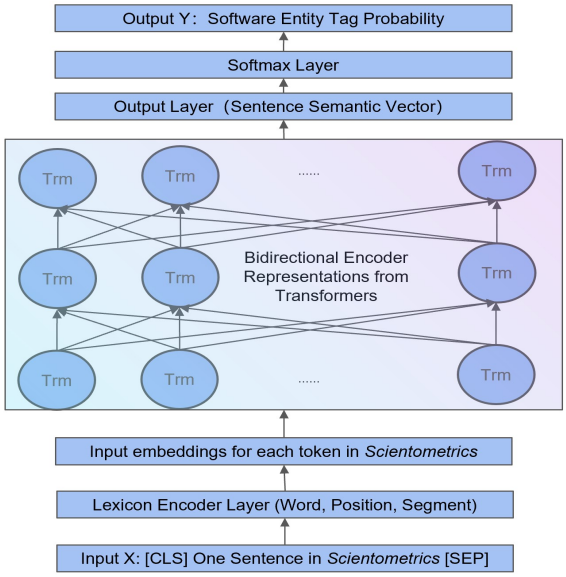

Fig. 8. Architecture of the BERT model for software entity extraction from *Scientometrics*

As shown in Fig. 2, CLS always appears at the start of the full-text of research articles from *Scientometrics* as a token of the sentence segmentation. The SEP token appears at the end of the research articles. Lexical coding is added on the three layers, namely, the word, word position, and sentence segmentation layers, to obtain the word embedding as the input. The multi-layer transformer based on the self-attention mechanism is then operated to generate the contextual semantic representation for each word in the full-text of the research articles from *Scientometrics*. Softmax is implemented through a neural network layer just before the output layer to predict the software entity tags and to lastly extract the software entities.

5 Experiments

5.1 Experimental Data Processing

In this study, the full text of the research articles from *Scientometrics* stored in the database were organised in paragraphs. As the maximum input length of the BERT model is 512, the Stanford NLP method (Manning et al. 2014) was implemented for the parsing of the paragraphs. The character length of each sentence was maintained under 512 in order for the text input to satisfy requirements of the model input. A total of 659,191 sentences with over 20 million characters was obtained after parsing. These sentences were then input into Brat for the text annotation (Stenetorp et al., 2012). The software entities in the documents were manually annotated, and 13,269 software entities were identified.

After the annotation, the textual data were processed in a format required by the sequence tagging model, and the BMES tagging scheme was used. In Fig. 3, the words are in the first column and the tags in the second column. B-Software, M-Software, and E-Software represent the beginning word of the software entity, middle word, and ending word, respectively. If a software entity was of one word, it was tagged as S-Software. The characters that did not denote software entities were tagged as O. The sentences were separated by a blank line as a separator.

Two	O
other	O
computer	O
programs	O
that	O
can	O
be	O
used	O
to	O
construct	O
graph-based	O
maps	O
are	O
CiteSpace	S-software
(	O
Chen	O
【	O
#CR7	O
】	O
)	O
and	O
the	O
Network	B-software
Workbench	M-software
Tool	E-software
.	O

Fig. 9. Example of the input of the sequence tagging model

5.2 Model Parameters and Experimental Environment

Various models, such as the CRF++, the BiLSTM-CRF, and BERT, were operated, and their software entity extraction performance was compared to identify which one has the best performance.

The CRF++ model is a discriminant probabilistic undirected graph model. In this study, the CRF++ 0.58 package (Kudo 2005) was used in combination with the basic feature template for software entity extraction.

The BiLSTM-CRF model consists of the embedding, bidirectional LSTM, and CRF layers. Gradient clipping was adopted to avoid gradient explosion and disappearance, with clip = 5.0 and the learning rate was set to 0.001. The dimensionality of the word embedding was 300. The number of hidden neurons in the LSTM layer was set to 256, and the Layer Num was set to 2. The batch size was 512. The number of training epochs was 200, and the Adam optimiser was used for the gradient descent optimisation. Early stopping was implemented to avoid overfitting and to accelerate the training speed, which implies that the training would be terminated if the F-value of the cross-validation set does not increase within 10 iterations.

The transformer is the primary component of BERT, a neural network model proposed by Google in 2018. Thus far, BERT has achieved great success in 11 NLP tasks. The output layer of the pre-trained BERT model in English was modified by transfer learning so that the BERT model was better suited for software entity extraction from the full-text of the research articles from *Scientometrics*. The number of hidden neurons was set to 768; the number of attention heads was 12; the warmup proportion was 0.3; the learning rate was 2.0E-5; the batch size was 16; the maximum sequence length was 512; and the training epoch was set to 3.

The training of a neural network typically involves a significant load of parallel computing and matrix calculation. Therefore, the throughput and the response speed cannot sufficiently meet the requirements if the deep learning process is performed on a CPU. In this study, the NVIDIA Tesla P40 GPU was used to train the neural network. This GPU delivers a data handling capacity, which is over 60 times higher than that of the CPU and has an inference capacity of up to 47 TOPS (tera operations per second). The configuration of the computer used in the experiment is as follows: CPU: 48 Intel(R) Xeon(R) CPUs E5-2650 v4 @ 2.20GHz; memory: 256 GB; GPU: 6 NVIDIA Tesla P40 cards; video memory: 24 GB; and operating system: CentOS 3.10.0.

5.3 Analysis of the Entity Extraction Performance

For the evaluation of the performance of model extraction software entity, it uses precision, recall and F1 value to measure the performance of software entity extraction. The calculation formula is shown in formula 10-12. The precision is the rate at which the software entities extracted by the model are correct, and the recall is the rate at which the software entities extracted by the model are extracted from the corpus. The F1 value is the reconciled weighted average of the recall and precision, which is used to measure the overall performance of the model recognition software entity.

$$\text{Precision}=\frac{\text{Number of software entities extracted correctly}}{\text{Total number of software entities extracted}}*100\% \quad (10)$$

$$\text{Recall}=\frac{\text{Number of software entities extracted correctly}}{\text{Total number of software entities in the corpus}}*100\% \quad (11)$$

$$\text{F1}=\frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}*100\% \quad (12)$$

Based on the corpus annotated above, the CRF, Bi-LSTM-CRF, and BERT models were operated for the software entity extraction experiments in the full-text of the research articles from *Scientometrics*. The 10-fold cross-validation process was implemented for each model to eliminate the influence of random errors on the experimental results. The corpus was split with a 9:1 ratio into a training set and a testing set. The averages of the precision, the recall, and F1 were calculated to measure and compare the overall entity extraction performance between the models.

Table 3. Comparison of the software entity extraction performance across the models

	CRF++			Bi-LSTM-CRF			BERT
No.	P	R	F1	P	R	F1	P
1	90.14%	65.93%	76.16%	80.63%	74.76%	77.59%	84.44%
2	90.94%	69.50%	78.79%	80.93%	78.61%	79.76%	86.72%
3	88.37%	68.89%	77.42%	84.71%	74.18%	79.09%	86.87%
4	91.90%	64.79%	76.00%	84.71%	74.18%	79.09%	87.13%
5	89.88%	67.85%	77.33%	83.74%	75.27%	79.28%	85.53%
6	90.76%	68.61%	78.15%	83.18%	76.67%	79.79%	86.70%
7	89.18%	66.38%	76.11%	82.08%	74.51%	78.11%	85.48%
8	91.18%	64.20%	75.35%	79.88%	76.63%	78.22%	87.45%
9	90.36%	70.30%	79.75%	82.70%	75.53%	78.90%	84.85%

As shown in Table 1, BERT exhibited the highest values of the average recall and F1 among the three models, with 83.89% and 84.99%, respectively. The CRF++ had the highest precision, which was 90.03%. The average value of F1 of the CRF++ model was 76.84%, which was lower than that of the Bi-LSTM-CRF and the BERT models by 1.83% and 8.15%, respectively. This result indicates that without adding the complex manual features, the CRF++ achieved a better recognition of high-frequency entities. Compared with the deep learning models, the CRF++ did not incorporate a neural network for the automatic feature extraction and lacked a semantic similarity mechanism, such as word embedding. Therefore, the CRF++ failed to recognise the semantically related software entities. Thus, the recall of the Bi-LSTM-CRF and BERT was higher than that of the CRF++ by 8.78% and 16.83%, respectively. The BERT model is pretrained with the large-scale unsupervised corpus by incorporating a transformer based on a self-attention mechanism. The output layer of the BERT model is modified with the help of transfer learning so that it is better suited for the software entity extraction from the full-text of the research articles from *Scientometrics*. Based on these advantages, it was observed that the BERT model exhibited a higher semantic modelling performance and the highest overall performance. The precision, the recall, and F1 of the BERT model were 86.13%, 83.89%, and 84.99%, respectively.

6 Conclusion

In this study, the full-text of research articles were collated from *Scientometrics* from 2010 to 2020. The corpus for the software entities in the full-text of the research articles was developed. The extracted software entities were further analysed from various perspectives, such as the distribution in the different parts of a document, number of software entity mentions and citations, and time-series evolution. It was observed that certain software entities were mentioned more frequently in Method and Experiment and Result sections. A general increasing trend was observed in the number of software entity mentions and citations in *Scientometrics* from 2010 to 2020. This

indicates that *Scientometrics* attached greater importance to the use of software when compared to power academic research and the standardised citations of software entities. A power law was observed in the correlation between the number of software mentions and the document number. The time-series analysis of the software entities showed that *Scientometrics* greatly favoured the Altmetrics research in the recent years, with an increase in the number of mentions of the relevant software products. New software entities for powering the research on new topics were also mentioned, such as ScientoPy and Google+.

The machine learning and the deep learning models, CRF, Bi-LSTM-CRF and BERT were established to extract the software entities from the research papers in *Scientometrics*. The highest precision (90.03%) was achieved with CRF++; the highest recall and F1-value of 83.89% and 84.99%, respectively, were achieved with BERT.

This study still faces certain limitations. For example, it only covered one journal, which was *Scientometrics*. In the future, the data sources will be expanded. The BERT-based model built in the present study must be applied for the software entity extraction from other journals to deepen the mining and the analysis of software entities from multiple perspectives.

Acknowledgements

Thank you very much to all the 25 graduate and undergraduate students who participated in the software entities annotation. The authors acknowledge the National Natural Science Foundation of China (Grant Numbers: 71974094) for financial support.

References

Bu, Y., Wang, B., Huang, W. B., Che, S., & Huang, Y. (2018). Using the appearance of citations in full text on author co-citation analysis. *Scientometrics*, *116*(1), 275-289.

Chen, H., & Luo, X. (2019). An automatic literature knowledge graph and reasoning network modeling framework based on ontology and natural language processing. *Advanced Engineering Informatics*, *42*, 100959.

Chen, Z., & Yan, E. (2017). Domain-independent term extraction & term network for scientific publications. *IConference 2017 Proceedings*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, Y., Song, M., Han, J., Yu, Q., Yan, E., Lin, L., & Chambers, T. (2013). Entitymetrics: Measuring the impact of entities. *PloS one*, *8*(8), e71416.

Glänzel, W., & Thijs, B. (2017). Using hybrid methods and 'core documents' for the representation of clusters and topics: the astronomy dataset. *Scientometrics*, *111*(2), 1071-1087.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Hu, Z., Lin, G., Sun, T., & Hou, H. (2017). Understanding multiply mentioned references. *Journal of Informetrics*, *11*(4), 948-958.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

Jeong, Y. K., Song, M., & Ding, Y. (2014). Content-based author co-citation analysis. *Journal of Informetrics*, *8*(1), 197-211.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Li, K., & Yan, E. (2018). Co-mention network of R packages: Scientific impact and clustering structure. *Journal of Informetrics*, *12*(1), 87-100.

Li, K., Rollins, J., & Yan, E. (2018). Web of Science use in published research and review papers 1997–2017: A selective, dynamic, cross-domain, content-based analysis. *Scientometrics*, *115*(1), 1-20.

Li, K., Yan, E., & Feng, Y. (2017). How is R cited in research outputs? Structure, impacts, and citation standard. *Journal of Informetrics*, *11*(4), 989-1002.

Lu, C., Bu, Y., Wang, J., Ding, Y., Torvik, V., Schnaars, M., & Zhang, C. (2019). Examining scientific writing styles from the perspective of linguistic complexity. *Journal of the Association for Information Science and Technology*, *70*(5), 462-475.

Lv, Y., Ding, Y., Song, M., & Duan, Z. (2018). Topology-driven trend analysis for drug discovery. *Journal of Informetrics*, *12*(3), 893-905.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014, June). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (pp. 55-60).

Nangia, U., & Katz, D. S. (2017, September). Track 1 paper: Surveying the US national postdoctoral association regarding software use and training in research. In *Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE5. 1)*.

Pan, X., Yan, E., Wang, Q., & Hua, W. (2015). Assessing the impact of software on science: A bootstrapped learning of software entities in full-text papers. *Journal of Informetrics*, *9*(4), 860-871.

Park, H., & Wolfram, D. (2019). Research software citation in the Data Citation Index: Current practices and implications for research software sharing and reuse. *Journal of Informetrics*, *13*(2), 574-582.

PASGRIMAUD, G. Pyquery: a jquery-like library for python, 2017.

Small, H. (2018). Characterizing highly cited method and non-method papers using citation contexts: The role of uncertainty. *Journal of Informetrics*, *12*(2), 461-480.

Small, H., Tseng, H., & Patek, M. (2017). Discovering discoveries: Identifying biomedical discoveries using citation contexts. *Journal of Informetrics*, *11*(1), 46-62.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., & Tsujii, J. I. (2012, April). BRAT: a web-based tool for NLP-assisted text annotation. *In Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (pp. 102-107).*

Song, M., Han, N. G., Kim, Y. H., Ding, Y., & Chambers, T. (2013). Discovering implicit entity relation with the gene-citation-gene network. *PloS one*, *8*(12), e84639.

Song, M., Heo, G. E., & Ding, Y. (2015). SemPathFinder: Semantic path analysis for discovering publicly unknown knowledge. *Journal of informetrics*, *9*(4), 686-703.

Thelwall, M. (2019). The rhetorical structure of science? A multidisciplinary analysis of article headings. *Journal of Informetrics*, *13*(2), 555-563.

Thijs, B., & Glänzel, W. (2018). The contribution of the lexical component in hybrid clustering, the case of four decades of "Scientometrics". *Scientometrics*, *115*(1), 21-33.

Wang, Y., & Zhang, C. (2018, March). Using full-text of research articles to analyze academic impact of algorithms. In *International Conference on Information* (pp. 395-401). Springer, Cham.

Yan, E., & Pan, X. (2015). A Bootstrapping Method to Assess Software Impact in Full-Text Papers. In ISSI.

Yan, E., & Zhu, Y. (2015). Identifying entities from scientific publications: A comparison of vocabulary-and model-based methods. *Journal of informetrics*, *9*(3), 455-465.

Yan, E., & Zhu, Y. (2018). Tracking word semantic change in biomedical literature. *International journal of medical informatics*, *109*, 76-86.

Yan, E., Williams, J., & Chen, Z. (2017). Understanding disciplinary vocabularies using a full-text enabled domain-independent term extraction approach. PloS one, 12(11), e0187762.

Yu, H., Xu, S., & Xiao, T. (2018). Is there Lingua Franca in informal scientific communication? Evidence from language distribution of scientific tweets. *Journal of Informetrics*, *12*(3), 605-617.

Yu, Q., Ding, Y., Song, M., Song, S., Liu, J., & Zhang, B. (2015). Tracing database usage: Detecting main paths in database link networks. *Journal of Informetrics*, *9*(1), 1-15.

Zhang, G., Ding, Y., & Milojević, S. (2013). Citation content analysis (CCA): A framework for syntactic and semantic analysis of citation content. *Journal of the American Society for Information Science and Technology*, *64*(7), 1490-1503.

Zhang, Y., Lu, J., Liu, F., Liu, Q., Porter, A., Chen, H., & Zhang, G. (2018). Does deep learning help topic extraction? A kernel k-means clustering method with word embedding. *Journal of Informetrics*, *12*(4), 1099-1117.

Zhao, M., Yan, E., & Li, K. (2018). Data set mentions and citations: A content analysis of full-text publications. *Journal of the Association for Information Science and Technology*, *69*(1), 32-46.