What2Cite: Unveiling Topics and Citations Dependencies for Scientific Literature Exploration and Recommendation

What2Cite: Unveiling Topics and Citations Dependencies for Scientific Literature Exploration and Recommendation

Davide Giosa, Luigi Di Caro

University of Turin, Turin, Italy
giosa.davide@gmail.com
luigi.dicaro@unito.it

Abstract. The continuous evolution of research has led to an exponential growth of the scientific literature. This engenders difficulties for researchers to entirely capture the most salient efforts related to their own research. In this paper, we propose a novel knowledge model for unveiling meaningful and labeled relations among articles based on both topics and latent citation dependencies. An experimentation on the whole literature in the Computer Science field allowed us to validate our approach by bridging the gap between few lines of textual content (e.g., an abstract) to the most relevant papers to be included in the bibliography.

Keywords: citation modeling · topic modelling · document semantic

1 Introduction

The search and the exploration of relevant information within large amounts of scientific papers is becoming more and more laborious. While there is an obvious connection between articles through the content that they express (i.e., their semantics), other dynamics related to the citational aspect of the scientific literature are also involved. In the light of this, in this work we aim at modeling relations among articles based on both their thematic information and the latent structure of the referenced citations within the entire literature of a generic domain (Computer Science, in our case).

Generally speaking, we started from the main goal of associating few lines of textual content (e.g., an abstract) with a set of papers that should be considered for inclusion as references. A first and standard view on this problem, which is indeed utilized by several online services, is that of providing (or suggest, recommend, etc.) articles which reflect similar content. However, it is known that a simple abstract may have an incredible large set of very similar documents, whereas the decision of what to include is completely left to the complex reasoning and knowledge of researchers, who may know highly deep details about fine-grained information and perspective contained in each single article. While we may consider such semantic depth as too complex to unravel for any Artificial Intelligence mechanism so far, we can utilize the *output* of such scientists' reasoning for going back up to it. This basis, in our view, is represented by the already-existing choice of references each article in literature carries within its bibliography. In other words, the co-occurrence of citations within the literature is an evidence of reasoning processes which relate the specific content of a paper to other articles irrespective of their surface lexical semantics (or topics).

When an article A is related to another article B, it usually happens when A and B share some features, possibly regardless of the similarity between their topics. For instance, article B may use the data of A although for different goals.

In this paper, we propose a citation-centered modeling approach which creates a semantic knowledge of both topics and clusters of citations (which we name *citopics* from now on) which allows a fair connection between a short textual summary with the most relevant references in the literature. Section 2 goes through the related works on the topic, while Section 3 presents the method[1] and its technicalities. Section 4 describes the evaluation of the proposal and Section 5 finally concludes the contribution highlighting critical points and possible future directions.

2 Related work

Our contribution has similarities with several approaches related to the modeling and use of the citations within large scholar databases, such as *(i)* semantic modeling of citations (e.g., [17, 12, 5]), *(ii)* data analysis and extraction of relevant information (e.g., [7, 15, 18, 6]), and *(iii)* exploration of the scientific literature by means of faceted search queries and visualization tools (e.g., [8, 9, 2, 13, 1]).

So far, part of the scientific literature is focused on the theoretical and top-down aspects of the citations. For example, *CiTo* [17] is focused on the modeling of possible citation intents. More in detail, they identified and formalized different types of possible citation meanings by proposing an ontology which includes a wide set of complex cases. However, this type of approach requires manual efforts of annotations and it is not suitable for large-scale analysis of a scientific domain.

On the other side, computational approaches to citation modeling have been presented, mostly based on clustering or classification tasks. In [10], the authors presented an unsupervised technique based on a clustering process, identifying and then manually labeling 11 classes of citations in a corpus. Differently, in [5] the authors proposed a classifier based on Scaffolds models [19] that was able to identify 6 classes of citations on the ACL-ARC dataset and 3 classes on a larger dataset named SciCite with high accuracy levels.

Our goal, conversely, is to focus on the association between the semantic content of papers and their bibliography, creating a dual space where to search for relevant candidate references to ascribe to short research descriptions.

[1] The code is publicly available at https://github.com/Dive904/What2Cite.

3 Data and Method

In this section we present the details of the proposed method and the utilized datasets and resources. We built our experimentation on the Semantic Scholar's dataset[2], filtering out non-English contributions[3], and reaching around 4 millions Computer Science papers associated with metadata such as *identifiers*, *years of publication*, *sources*, *titles*, *abstracts* and *out citations*.

3.1 Topic Modeling and Classification

The first phase of the approach consists in the extraction of the topics from the textual abstracts contained in the dataset. To this end, we employed the well known Latent Dirichlet Allocation (LDA) technique[4] [4] on half of the dataset, only considering articles from 2010 and applying standard text preprocessing steps such as lemmatization and stopwords removal[5].

We experimented with different numbers of topics, finally focusing on 35, 40 and 45. Then, after a careful qualitative analysis on the obtained results, we decided to opt for 40 topics. Below, we show some examples:

```
Topic #3: security attack scheme privacy protocol key secure
Topic #8: gene available tool analysis database file sequence
Topic #13: problem algorithm fuzzy solution set time optimization
Topic #27: patient medical health clinical disease treatment care
Topic #32: text word document task information semantic topic
```

Generally speaking, we consider the whole set of 40 topics as of very high quality, with high topic coherence and consistency.

After this phase, we used the trained LDA model to further train a Neural Network model for classifying new instances. In particular, we created a dedicated dataset by picking up 2000 random papers which were most highly-associated with each topic, then dividing it as follows: 60% for the training set, 30% for the test set, 10% for the validation set.

The employed Neural Network model is a Bidirectional LSTM [16]. This allows us the classification of new textual contents (e.g., new abstracts) based on stable LDA topics, also taking into account the sequential nature of the natural language and the recent advancement of neural-based word embedding technologies. The overall model architecture is shown in Fig. 1, and it is composed of a first embedding layer with 300 dimensions and a Bidirectional LSTM of 550 units, followed by a dropout of 0.4. We used GloVe embeddings [14] trained with 840 billion of entities and 2.2 million of words, with a vector size of 300.

Figure 2 shows the model accuracy during the training. We trained the network for 20 epochs, and selected the epoch 17 for the final model, reaching the accuracy scores shown in Table 1.

[2] Dataset available at https://api.semanticscholar.org/corpus/download/

[3] *langdetect* - https://pypi.org/project/langdetect/

[4] *scikit-learn* - https://scikit-learn.org/0.16/modules/generated/sklearn.lda.LDA.html

[5] *nltk* - https://www.nltk.org.

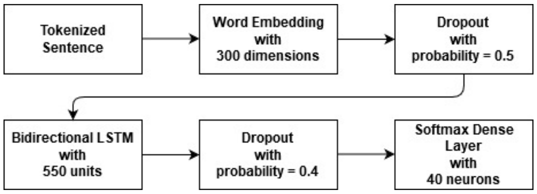

Tokenized Sentence
Word Embedding with 300 dimensions
Dropout with probability = 0.5
Bidirectional LSTM with 550 units
Dropout with probability = 0.4
Softmax Dense Layer with 40 neurons

Fig. 1: The proposed Neural Network architecture.

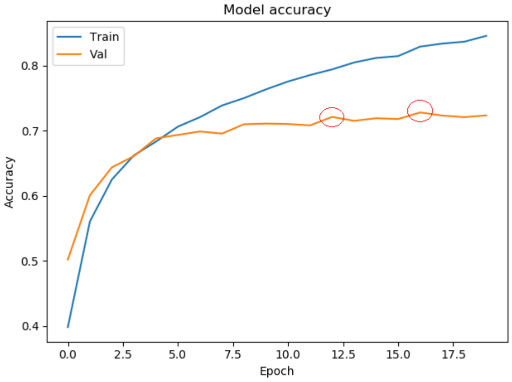


Fig. 2: Model Accuracy during training.

3.2 Citation Topic Modeling: the concept of *Citopics*

A *Citopic* (a.k.a. *Citation Topic*) is a set of paper IDs which, within the literature, are most often cited together. Thus, we considered this scenario as fitting a standard application of Topic Modeling where, instead of words, the input documents are composed of citations (IDs of papers). By treating every citation as a single word, a single bibliography of an input paper is transformed in a sentence-like sequence of cited papers where to apply a topic modeling exercise. In this particular case, the output is a set of topics (that we rename as *citopics*) containing paper IDs often cited together in the literature.

Since the dataset has some missing *OutCitations* information, we first made a scan of the entire dataset creating a new version where every paper has at least $\frac{2}{3}$ of its citations in the dataset. This new dataset counts approximately 800K papers. As in the first (thematic) Topic Modeling on the abstracts, we run the LDA model on the citations trying different numbers of topics n. Due to the goal of unraveling many small sets of papers, we maximized n, reaching the value of $n = 750$ before encountering technical problems with the used library. After obtaining 750 different citopics, we applied a *rank-and-filter* approach. In particular, we sorted the citopics words (i.e., citations) and kept those until

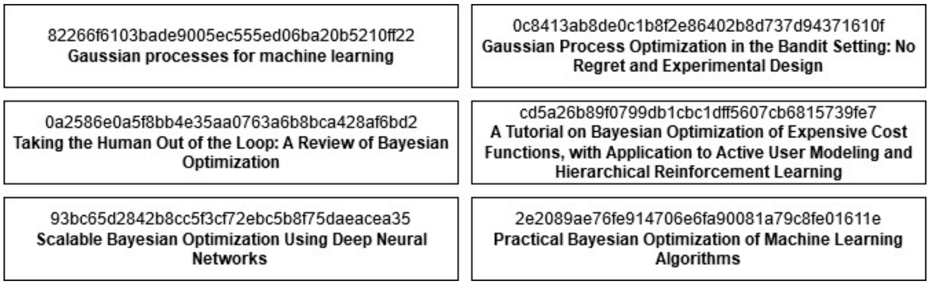
Table 1: Model scores at epoch 17.

Training		Validation		Test	
Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
2.0715	0.8292	2.4107	0.7280	1.1776	0.7197

Table 2: Model scores with top 3 elements and using threshold

Set	Top 3	Using Threshold of 0.4
Training	0.922	0.735
Validation	0.98	0.893
Test	0.918	0.724

summing up to the $\frac{2}{3}$ of the entire citopics scores[6]. Figure 3 shows a citopic example.



82266f6103bade9005ec555ed06ba20b5210ff22
Gaussian processes for machine learning

0c8413ab8de0c1b8f2e86402b8d737d94371610f
Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design

0a2586e0a5f8bb4e35aa0763a6b8bca428af6bd2
Taking the Human Out of the Loop: A Review of Bayesian Optimization

cd5a26b89f0799db1cbc1dff5607cb6815739fe7
A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning

93bc65d2842b8cc5f3cf72ebc5b8f75daeacea35
Scalable Bayesian Optimization Using Deep Neural Networks

2e2089ae76fe914706e6fa90081a79c8fe01611e
Practical Bayesian Optimization of Machine Learning Algorithms

Fig. 3: An example of citation topic (*Citopic*), mostly belonging to Topic 38 (*algorithm search problem* ...) and Topic 1 (*network feature learning* ...).

3.3 Linking Topics and Citopics: the $L_t(x)$ function

To relate *citopics* with *topics*, we define a function $L_t(x) : String \rightarrow List$ (where x is a paper ID) which returns a list of t elements, each one identifying one topic. In detail, each element i reports the number of times the paper x is cited by a paper of topic i. Algorithm 1 illustrates this dictionary creation process.

In our case, we can use the function $L_{40}(x)$, where x is a paper ID in one *citopic*, to get the list of frequencies. To make an example, we can call this function with a sample paper *id*:

$$\texttt{L\_EXP} = L_{40}(id)$$

[6] The score of a citopic is the sum of all the scores of a single ID in the citopic. We can get this score from the LDA while creating the clusters.

Algorithm 1 Create Dictionary

```
$D \leftarrow dataset$
$Dic \leftarrow initialize\_dictionary()$
**for all** paper $p$ in $D$ **do**
   $id \leftarrow get\_paper\_id(p, D)$
   $topic \leftarrow get\_paper\_topic(p, D)$
   $outcitations \leftarrow get\_paper\_outcitations(p, D)$
   **for all** citation $c$ in $outcitations$ **do**
      $score\_list \leftarrow Dic.get(c)$
      $score\_list[topic] \leftarrow score\_list[topic] + 1$
      $Dic[c] \leftarrow score\_list$
   **end for**
**end for**
```

from which we get the following list:

```
[0, 36, 1, 0, 0, 0, 0, 0, 0, 1, 3, 0, 0, 1, 0, 0, 0, 0, 1, 0,
 1, 0, 0, 1, 2, 0, 0, 0, 0, 1, 0, 106, 0, 0, 0, 0, 2, 61, 1, 0]
```

As we can notice, the result is a list of size 40 where every element is a number (e.g., `L_EXP[31] = 106`, meaning that the input paper has been cited 106 times by another paper of topic #31).

3.4 Linking Texts to Citopics: the $ScoreCitTopic_c(x)$ function

Now, we can define a function that recommends *citopics* given an input abstract of a generic paper x. Let us define the $ScoreCitTopic_c(x) : String \rightarrow Int$ function with Algorithm 2.

Algorithm 2 Score Function

```
$cit \leftarrow cit\_topic$
$x \leftarrow paper$
$topic \leftarrow get\_paper\_topic(x)$
$final\_score \leftarrow 0$
**for all** $id$ in $cit$ **do**
   $score\_list \leftarrow L_{40}(id)$
   $score \leftarrow score\_list[topic]$
   $final\_score \leftarrow final\_score + score$
**end for**
**return** $final\_score$
```

The score function is based on the $L_t(x)$ function. We start from the topic of the input paper and we scan the input *citopic*. First, for each paper within the *citopic*, we obtain the frequency list through the $L_t(x)$ function. At this point, we then take the frequency related to the topic associated with the input paper x. In words, with this step we are answering the following question: “how many times a paper in the *citopic* has been cited by papers with the same topic of the input paper?”. By answering this question for each paper in *citopic* we can obtain a global score. After this phase, we take the score of every available *citopic* for a specific paper.

Algorithm 3 Score All Citopics Function

```
$x \leftarrow paper$
$cit\_topics \leftarrow get\_cit\_topics()$
$score\_list \leftarrow empty\_list()$
for all $cit$ in $cit\_topics$ do
    $score \leftarrow ScoreCitTopic_{cit}(x)$
    $score\_list.append(score)$
end for
return $score\_list$
```

In particular, Algorithm 3 returns a list with the same size of the total number of *citopics*, that is, in our case, 750. Using this function with a particular input paper x, we obtain the following list $[s_0, s_1, \ldots, s_{749}]$. Here, the generic element s_i represents the score for the i-th *citopic*. We then conclude with Algorithm 3 that relates a particular paper to different *citopics*, in particular by assigning a score to each of them. An immediate usage of this result is the possibility of sorting the list to obtain the final articles recommendation. In detail, from the list in 3.4, we can create a second list as follows: $[(0, s_0), (1, s_1), \ldots, (749, s_{749})]$, where every element is a pair (i, s_i). Now, by sorting the pairs through the scores s_i we reach a rank of *citopics* indices. From this, we may select the first k *citopics* with highest scores for recommendation. The overall architecture of the proposal is shown in Figure 4.

4 Evaluation

4.1 Miss Set and Hit Set

Let us consider a paper x with a list O of out citations: $O = [o_0, \ldots, o_n]$, where each o_i is a paper cited by paper x. Similarly, if we pick up a *citopic* C, we have $C = [c_0, \ldots, c_m]$, where each c_j is a paper in that cluster. From these sets we can generate an H-set, also called *Hit Set*, in the following way: $H = C \cap O$, where H represents intersection between C and O. With the H-set, we can create the *Miss set* M, i.e. $M = C - H$. If we consider a function $out(x)$ which returns the *out citation* set of a paper x, we can define a function $hit_C(x)$, where $hit_C(x) = C \cap out(x)$. The $hit_C(x)$ function of a paper x with a *citopic* C returns a set containing all the papers within the *out citation* set of x that are actually in *citopic* C. Thus, it returns the H-set of a paper x with a *citopic* C.

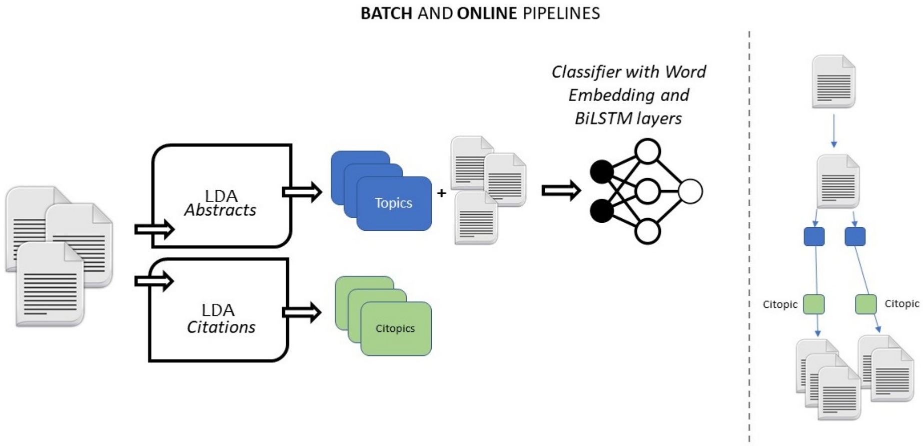


Fig. 4: Batch and online architecture of the proposal.

4.2 Accuracy definition

We need to remember that the *citopics* are generated from an LDA run. Thus, it may happen that a paper ID does not appear in any *citopic*. Thus, to create a fair evaluation of our proposed method, we counted its obtained *hits* in relation with the total citations that have been covered by all *citopics* only.

Let us consider a function $ht(x)$ from where we can enumerate all the possible hits for a specific paper x. So, we can calculate the accuracy with:

$$A_C(x) = \frac{|hit_C(x)|}{ht(x)} \tag{1}$$

We can make things a little more complex, considering both different *citopics* and different papers at the same time. In particular, we may have a set of papers $P = [p_0, \ldots, p_n]$ and a set of different *citopics* for a particular paper p_i $C^{p_i} = [C_0^{p_i}, \ldots, C_m^{p_i}]$ where $C_j^{p_i}$ is a selected *citopic* for paper p_i. We can then suppose to make another set $C = [C^{p_0}, \ldots, C^{p_n}]$.

Finally, the general accuracy formula becomes:

$$A_C(P) = \frac{\sum_{p_i \in P} \sum_{C_j^{p_i} \in C^{p_i}} |H_{C_j^{p_i}}(p_i)|}{\sum_{p_i \in P} HT(p_i)} \tag{2}$$

In the numerator, the first sum loops on each input paper p_i, while the second one is used to loop on each *Citopic* $C_j^{p_i}$ in the selected list C^{p_i}. Then, we use the H function with $C_j^{p_i}$ and p_i as input, to calculate the *Hit Set* and, in particular, its cardinality. In the denominator, we have a single loop on every input paper p_i where we calculate the number of all possible hits that could be obtained. In conclusion, the $A_C(P)$ function calculates the percentage of obtained hits out of the total possible hits. Algorithm 4 shows the accuracy calculation process.

Algorithm 4 Accuracy with more papers and more *Citopics*

```
$P \leftarrow list\_of\_papers$
$n \leftarrow input\_number$
$C \leftarrow empty\_list()$
for all $paper$ in $P$ do
    $scores \leftarrow AllScoreFunction(paper)$
    $scores \leftarrow sort(scores)$
    $citopics \leftarrow get\_citopics\_with\_max\_scores(n)$
    $C.append(citopics)$
end for
$acc \leftarrow A_C(P)$
return $acc$
```

To have a precise idea on the accuracy of the entire system, we could use Algorithm 4 and with the n parameter equals to 750. In this way, we take all the *citopics* and calculate the accuracy on each individual *citopic*. The idea is that if the individual accuracy values calculated on each *citopic* have a decreasing trend as the scores drop, then the result may be considered as satisfactory.

In detail, let us imagine a simple score list $L = [s_0, \dots, s_{749}]$. Then, we divide the scores list into further sublists (or portions). Taking as input a certain percentage p, we have to divide the list of scores into sublists composed of p percent of the total elements. If the percentage is 10%, each individual sublist will be composed of 10% of the elements (data binning). This way, we do not take into account top-ranked elements but top-ranked portions. Thus, the list will be $L = [[s_0, \dots, s_{49}], \dots, [s_{674}, \dots, s_{749}]]$. With this combination of sublists, we can take a single sublist, then the *citopics* related to that sublist[7], and compute the cardinality of the Hit Set for each *citopic*. Finally, we can calculate the sum over all the citopics. Thus, if the percentage is e.g. 10%, we should have a partitioned list $PL = [l_0, \dots, l_9]$ where every l_i is the number of hits from the *citopics* in portion i of the list L.

4.3 Results

We calculated the accuracy of the proposed method on a random set of 1000 input papers. It must be specified that the papers taken as input are papers that have not been used for the creation of the *citopics* (i.e., it can be considered as a test test). In this way, we also simulate the behaviour of our approach in the real case in which the input is a paper that does not yet exist in the literature. During this process, we take into account all the classified topics of a single paper with a probability higher than 40%. Results are shown in Figure 5.

The decreasing trend of hits percentage ($P1$ = 65.5%, $P2$ = 14.9%, $P3$ = 7.1%, $P4$ = 4.3%, etc.) is a clear sign that the score-based metric is the right one, since we see a drop in the number of hits together with the decreasing of the *citopic* scores. This trend leaves complete freedom to the user (or application)

[7] Every score s_i is related to the *Citopic* i.

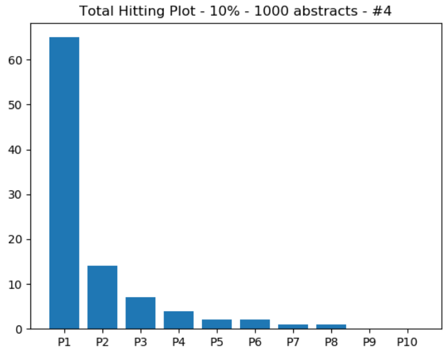


Fig. 5: Cumulative distribution of hits over the portions.

regarding the choice of the number of *citopics* to consider. For example, taking the first 2 portions and then the 4 portions, the method cumulatively reach around 80% and 91% of accuracy respectively (i.e., probability that the returned papers fit the input textual content or abstract).

5 Conclusions and Future works

Linking an abstract to single citations can be very challenging, as the simple content-based similarity may end up with thousands of equally-relevant articles. For this reason, in this paper we proposed a method which includes information about citations dependencies through a topic modeling techniques applied on paper IDs, obtaining very promising results. At the moment, we based our efforts on some qualitative analysis (e.g., the choice of the number of topics) which can be certainly improved in future research. Another type of extension could be based on hierarchical topic modeling [11]. Indeed, topics are often correlated and standard topic modeling techniques are not able to capture these relationships [3]. Another future direction could focus on the used metrics for the accuracy evaluation. In this work, only frequencies have been used, whereas other types of statistical information might be employed.

References

1. Akujuobi, U., Zhang, X.: Delve: A dataset-driven scholarly search and analysis system. SIGKDD Explor. Newsl. **19**(2), 36–46 (Nov 2017). https://doi.org/10.1145/3166054.3166059, http://doi.acm.org/10.1145/3166054.3166059
2. Alexander, E., Kohlmann, J., Valenza, R., Witmore, M., Gleicher, M.: Serendip: Topic model-driven visual exploration of text corpora. In: Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on. pp. 173–182. IEEE (2014)

3. Blei, D.M., Lafferty, J.D., et al.: A correlated topic model of science. The Annals of Applied Statistics **1**(1), 17–35 (2007)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (Mar 2003)
5. Cohan, A., Ammar, W., van Zuylen, M., Cady, F.: Structural scaffolds for citation intent classification in scientific publications. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 3586–3596 (2019), https://www.aclweb.org/anthology/N19-1361/
6. Di Caro, L., Cataldi, M., Schifanella, C.: The d-index: Discovering dependences among scientific collaborators from their bibliographic data records. Scientometrics **93**(3), 583–607 (2012)
7. Šubelj, Nees Jan van Eck, Ludo Waltman, L.: Clustering Scientific Publications Based on Citation Relations: A Systematic Comparison of Different Methods. PLoS ONE 11(4) (2016)
8. van Eck, Ludo Waltman, N.J.: VOS: a new method for visualizing similarities between objects. Advances in Data Analysis: Proceedings of the 30th Annual Conference of the German Classification Society (pp. 299-306). Springer (2007)
9. van Eck, Ludo Waltman, N.J.: CitNetExplorer: A new software tool for analyzing and visualizing citation networks. Journal of Informetrics, 8(4), 802-823 (2014)
10. Ferrod, R., Schifanella, C., Caro, L.D., Cataldi, M.: Disclosing citation meanings for augmented research retrieval and exploration. In: The Semantic Web - 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2-6, 2019, Proceedings. pp. 101–115 (2019). https://doi.org/10.1007/978-3-030-21348-0_7, https://doi.org/10.1007/978-3-030-21348-0_7
11. Griffiths, T.L., Jordan, M.I., Tenenbaum, J.B., Blei, D.M.: Hierarchical topic models and the nested chinese restaurant process. In: Advances in neural information processing systems. pp. 17–24 (2004)
12. Kim, J., Kim, D., Oh, A.: Joint modeling of topics, citations, and topical authority in academic corpora. arXiv preprint arXiv:1706.00593 (2017)
13. Nagwani, N.: Summarizing large text collection using topic modeling and clustering based on mapreduce framework. Journal of Big Data **2**(1), 6 (2015)
14. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
15. Popescul, A., Ungar, L.H., Flake, G.W., Lawrence, S., Giles, C.L.: Clustering and identifying temporal trends in document databases. In: adl. p. 173. IEEE (2000)
16. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
17. Shotton, S.P.D.: FaBiO and CiTO: ontologies for describing bibliographic resources and citations. Web Semantics: Science, Services and Agents on the World Wide Web. Volume 17, Pages 33-43 (2012)
18. Strapparava, R.M.C.C.C.: Corpus-based and knowledge-based measures of text semantic similarity. AAAI'06 Proceedings of the 21st national conference on Artificial intelligence, Volume 1, Pages 775-780 (2006)
19. Swayamdipta, S., Thomson, S., Lee, K., Zettlemoyer, L., Dyer, C., Smith, N.A.: Syntactic scaffolds for semantic structures. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. pp. 3772–3782 (2018), https://www.aclweb.org/anthology/D18-1412/