WOSP 2020

Proceedings of the 8th International Workshop on Mining Scientific Publications

05 August, 2020
Wuhan,
China

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-952148-88-0

Introduction

The entire body of research literature is currently estimated at 100-150 million publications, with an annual increase of around 1.5 million. Research literature constitutes the complete representation of knowledge we have assembled as human species. It enables us to develop cures to diseases, solve challenging engineering problems and answer many of the world's challenges we are facing today. Systematically reading and analysing the full body of knowledge is now beyond the capacities of any human being. Consequently, it is essential to understand better how we can leverage Natural Language Processing/Text Mining techniques to aid knowledge creation and improve the process by which research is being done.

This workshop aims to bring together people from different backgrounds who:

1. have experience with analysing and mining databases of scientific publications,
2. develop systems that enable such analysis and mining of scientific databases or
3. who develop novel technologies that improve the way research is being done.

The topics of the workshop were organised around the following themes:

1. The whole ecosystem of infrastructures including repositories, aggregators, text-and data-mining facilities, impact monitoring tools, datasets, services and APIs that enable analysis of large volumes of scientific publications.
2. Semantic enrichment of scientific publications utilising text and data mining.
3. analysis of large databases of scientific publications to identify research trends and improve access to research content.

This year, we hosted a new shared task:

3C Citation Context Classification Shared Task

Recent years have witnessed a massive increase in the amount of scientific literature and research data being published online, providing revelation about the advancements in the field of different domains. The introduction of aggregator services like CORE has enabled unprecedented levels of open access to scholarly publications. The availability of full text of the research documents facilitates the possibility of extending the bibliometric studies by identifying the context of the citations. The shared task organized as part of the WOSP 2020 focused on classifying citation context in research publications based on their influence and purpose.

- **Subtask A:** Multiclass classification of citations into one of six classes: Background, Uses, Compare_Contrast, Motivation, Extension and Future.
- **Subtask B:** Binary classification of citations based on the classes Incidental and Influential, a task for identifying the importance of a citation.

Given a citation context, the participants were required to predict the intent of the citations. The participants were provided with a labelled dataset of 3000 training instances annotated using the ACT platform.

We are grateful to the program committee for their careful and thoughtful reviews of the submitted papers. Likewise, we are thankful to the keynote speakers for sharing their research and their vision for the field, and to the workshop attendees for a lively and productive discussion.

Petr Knoth
Christopher Stahl
Bikash Gyawali
David Pride
Suchetha N. Kunnath
Drahomira Herrmannova

Committees

Organisers:

Petr Knoth, Knowledge Media institute, The Open University, UK
Christopher Stahl, Oak Ridge National Laboratory, USA
Bikash Gyawali, Knowledge Media institute, The Open University, UK
David Pride, Knowledge Media institute, The Open University, UK
Suchetha N. Kunnath, Knowledge Media institute, The Open University, UK
Drahomira Herrmannova, Oak Ridge National Laboratory, USA

Program Committee:

Akiko Aizawa, National Insutitute of Informatics, Japan
Iana Atanassova , Université de Bourgogne Franche-Comté, France
Marc Bertin, Université Claude Bernard Lyon 1, France
José Borbinha, Universidade de Lisboa, Portugal
Pravallika Devineni, Oak Ridge National Laboratory, USA
Tirthankar Ghosal, Indian Institute of Technology Patna, India
Saeed-Ul Hassan, Information Technology University, Pakistan
Radim Hladik, Czech Academy of Sciences, Czech Republic
Monica Ihli, University of Tennessee, USA
Roman Kern, Graz University of Technology, Austria
Martin Klein, Los Alamos National Laboratory, USA
Birger Larsen, Aalborg University, Denmark
Paolo Manghi, ISTI-CNR, Italy
Sepideh Mesbah, Delft University of Technology, Netherlands
Peter Mutschke, GESIS Leibniz Institute for the Social Sciences, Germany
Federico Nanni, Alan Turing Institute, UK
Francesco Osborne, The Open University, UK
Robert M. Patton, Oak Ridge National Laboratory, USA
Eloy Rodrigues, Universidade do Minho, Portugal
Wojtek Sylwestrzak, ICM Univeristy of Warsaw, Poland
Vetle Torvik, University of Illinois, USA
Jian Wu, Old Dominion University, USA

Invited Speakers:

Anne Lauscher, University of Mannheim
Allan Hanbury, Vienna University of Technology
David Jurgens, University of Michigan
Neil Smalheiser, University of Illinois at Chicago
Kuansang Wang, MSR Outreach Academic Services

Table of Contents

Virtual Citation Proximity (VCP): Empowering Document Recommender Systems by Learning a Hypothetical In-Text Citation-Proximity Metric for Uncited Documents 1
Paul Molloy, Joeran Beel and Akiko Aizawa

Citations Beyond Self Citations: Identifying Authors, Affiliations, and Nationalities in Scientific Papers 9
Yoshitomo Matsubara and Sameer Singh

SmartCiteCon: Implicit Citation Context Extraction from Academic Literature Using Supervised Learning 21
Chenrui Guo, Haoran Cui, Li Zhang, Jiamin Wang, Wei Lu and Jian Wu

Synthetic vs. Real Reference Strings for Citation Parsing, and the Importance of Re-training and Out-Of-Sample Data for Meaningful Evaluations: Experiments with GROBID, GIANT and CORA 27
Mark Grennan and Joeran Beel

Term-Recency for TF-IDF, BM25 and USE Term Weighting 36
Divyanshu Marwah and Joeran Beel

The Normalized Impact Index for Keywords in Scholarly Papers to Detect Subtle Research Topics 42
Daisuke Ikeda, Yuta Taniguchi and Kazunori Koga

Representing and Reconstructing PhySH: Which Embedding Competent? 48
Xiaoli Chen and Zhixiong Zhang

Combining Representations For Effective Citation Classification 54
Claudio Moisés Valiense de Andrade and Marcos André Gonçalves

Scubed at 3C task A - A simple baseline for citation context purpose classification 59
Shubhanshu Mishra and Sudhanshu Mishra

Scubed at 3C task B - A simple baseline for citation context influence classification 65
Shubhanshu Mishra and Sudhanshu Mishra

Amrita_CEN_NLP @ WOSP 3C Citation Context Classification Task 71
Premjith B and Soman Kp

Overview of the 2020 WOSP 3C Citation Context Classification Task 75
Suchetha N. Kunnath, David Pride, Bikash Gyawali and Petr Knoth

Virtual Citation Proximity (VCP): Empowering Document Recommender Systems by Learning a Hypothetical In-Text Citation-Proximity Metric for Uncited Documents

Paul Molley
Trinity College Dublin
School of Computer Science and Statistics, Ireland
molloyp1@tcd.ie

Joeran Beel *
University of Siegen
Dept. of Electrical Engr. & Computer Science
Germany
joeran.beel@uni-siegen.de

Akiko Aizawa
National Institute of Informatics (NII), Digital Content and Media Sciences
Tokyo, Japan
aizawa@nii.ac.jp

Abstract

The relatedness of research articles, patents, court rulings, web pages, and other document types is often calculated with citation or hyperlink-based approaches like co-citation (proximity) analysis. The main limitation of citation-based approaches is that they cannot be used for documents that receive little or no citations. We propose Virtual Citation Proximity (VCP), a Siamese Neural Network architecture, which combines the advantages of co-citation proximity analysis (diverse notions of relatedness / high recommendation performance), with the advantage of content-based filtering (high coverage). VCP is trained on a corpus of documents with textual features, and with real citation proximity as ground truth. VCP then predicts for any two documents, based on their title and abstract, in what proximity the two documents would be co-cited, if they were indeed co-cited. The prediction can be used in the same way as real citation proximity to calculate document relatedness, even for uncited documents. In our evaluation with 2 million co-citations from Wikipedia articles, VCP achieves an MAE of 0.0055, i.e. an improvement of 20% over the baseline, though the learning curve suggests that more work is needed.

1 Introduction

Calculating document relatedness is key in creating recommender systems for digital libraries (we focus on research paper recommenders – our work is, however, equally applicable to patents, websites, court rulings and other documents with hyperlinks, citations respectively). Recommender systems in digital libraries calculate relatedness of research articles typically via content-based filtering or hyperlink/citation-based approaches (Jannach et al., 2010; Beel et al., 2016; Lops et al., 2019). Citation-based approaches consider documents as related that reference the same documents (bibliographic coupling), that are co-cited by other documents or that are otherwise connected in the citation graph (Beel et al., 2016).

Citation-based approaches may recommend more diverse items than content-based filtering, as citations can be made for various reasons (Willett, 2013; Färber and Sampath, 2019; Erikson and Erlandson, 2014). For instance, two documents can be co-cited because they address the same research problem; use the same methodology (to solve different problems); or two documents may be co-cited for less predictable reasons. Today's text-based methods can hardly distinguish such diverse types of relatedness. Instead, text-based methods generally consider two documents as related the more terms they have in common [1].

A particularly promising citation-based approach is Citation Proximity Analysis (CPA) (Gipp and Beel, 2009), which is illustrated in Figure 1. CPA considers documents as the more related, the closer the distance in which they are co-cited. For instance, in the example, the *Citing Document* cites *Document A* and *Document B* in the same sentence. *Document C* is cited in a different paragraph. Hence, A and B are more related than A and C (or B and C).

CPA out-performs standard co-citation analysis by up to 95% (Schwarzer et al., 2016) and has successfully been used with research articles (Balaji et al., 2017; Liu and Chen, 2011; Knoth and Khadka, 2017; Gipp and Beel, 2009), Wikipedia (Schwarzer et al., 2016, 2017), web pages (Gipp et al., 2010), mind-maps (Beel and Gipp, 2010) and authors (Kim et al., 2016). The downside of CPA is that it can be only be applied to documents that are (co-)cited. Most research articles, however, are

[1] Of course, there are multiple approaches like word embeddings that go beyond a simple term-overlap comparison. However, eventually, text-based approaches focus on content similarity, which is just one type of relatedness.

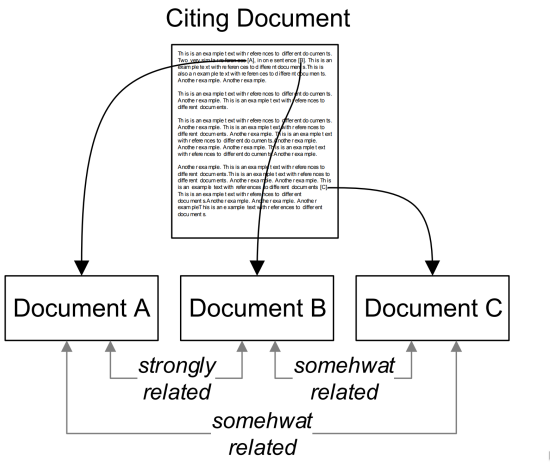


Figure 1: Illustration of Citation Proximity Analysis (Gipp and Beel, 2009). A citing document cites the three documents A, B, and C. Documents A and B are cited within the same sentence and are hence strongly related. Documents A and C, as well as documents B and C, are each cited within different paragraphs. Hence, they are considered as less strongly related to each other. A recommender system that receives document B as input, and that should recommend the most related document, would recommend document A.

never cited, and even if they are, it usually takes a year or more before they receive their first citation (Golosovsky, 2017; Abramo et al., 2016). Consequently, CPA has a low coverage, i.e. it can only be applied to a small fraction of research articles in a corpus and only relatively late.

We propose[2], implement and evaluate a novel approach that we name 'Virtual Citation Proximity' **(VCP)**. We hypothesize that VCP combines the advantages of co-citation proximity analysis (diverse notions of relatedness / high recommendation effectiveness), with the advantage of content-based filtering (high coverage). Hence, we expect that VCP advances the state-of-the-art in related-document calculations for search engines and recommender systems significantly.

2 Virtual Citation Proximity (VCP)

Virtual Citation Proximity (VCP) predicts in which distance two documents – that are not co-cited – would be co-cited if they were co-cited. This predicted proximity can then be used in the same way as real co-citation proximity to calculate document relatedness. At an abstract level, the idea behind VCP is that there is an inherent concept of relatedness between articles. This inherent relatedness can be described either through text or co-citations. As both, text and citations, eventually refer to the same relatedness, the text and citation are kind of a 'siamese twin'.

We propose to implement VCP via artificial neural networks that are trained with textual features – e.g. terms or word embeddings from the title or abstract – as input, and real citation proximity as target. In other words, we feed a neural network with pairs of documents of which we know how strongly they are related (expressed by the real proximity of their co-citations). The network then learns a similarity function that predicts based on the text the degree to which the two documents are related – even if the two documents have no terms or word embeddings in common.

We hypothesize that a neural network will be able to learn the diverse types of relatedness inherent to co-citations. Once the network is trained, it receives the text of two documents as input, and predicts in what proximity these two documents would be co-cited if they were co-cited. VCP can be applied to all document pairs in a corpus that contain a title (and abstract), i.e. typically all document in a corpus (100% coverage). If the predictions of VCP are precise, a recommender system based on VCP would be as effective as a system based on real citation proximity, but with a coverage as high as content-based filtering (100%).

Although Virtual Citation Proximity is based on textual features as input, we hypothesise that VCP will create recommendations similar to those based on real citation-proximity, since the machine learning algorithm is trained on real citation proximity as ground truth. With the recent advances in (deep) machine learning we hypothesise that a (deep) machine-learning algorithm will be able to detect hidden layers in the text. These will allow determining what makes two documents related, more reliable than the typical assumption in text-based approaches that two documents are related when they share the same terms or embeddings.

3 Related Work

Virtual Citation Proximity trains a machine learning model with real citation proximity as ground

[2]We proposed VCP previously in a non-peer-reviewed research proposal, but did neither implement nor evaluate it (Beel, 2017). Also, please note that the work we present is based on Paul Molloy's Bachelor thesis "Virtual Citation Proximity: Using Citation-Ground Truth to Train a Text-Based Machine Learning Model" at Trinity College Dublin, Ireland, 2018/2019. The Bachelor thesis is not (yet) published.

Figure 2: Screenshot of the MeSH classification tree

truth / target, and to the best of our knowledge we are first to do this. The method that is closest to using citation-proximity as ground truth for machine learning is using expert judgements (or knowledge bases) as ground truth, e.g. MeSH, ACM CCS, or DMOZ (Mohammadi et al., 2016; Hassan, 2017).

For instance, the MeSH classification is a classification tree that represents the major fields and subfields in the biomedical domain. MeSH was created by medical experts and biomedical manuscripts are often classified with MeSH, i.e. manuscripts are assigned to one of the MeSH categories, whereas two documents in the same category are considered to be related, and can be used either for training machine learning models or evaluating recommendation approaches (Hassan, 2017). Machine learning algorithms can infer from the existing documents in a category, which textual features make a document likely to belong to a certain category. New documents can then automatically be classified based on their text (Peng et al., 2018),

There are disadvantages to using expert classifications like MeSH, when compared to citations and VCP respectively. First, expert classifications are often one-dimensional, i.e. they provide only one type of relatedness (typically, the overall topic a research article is about). Second, most expert classification schemes allow documents to be in few categories only, and they focus on one field (e.g. medicine *or* computer science). Especially with today's increasingly interdisciplinary work, this is often not enough to adequately find all related documents. Third, classification schemes typically have a limited number of categories (a few thousand at most). This means, in large collections, categories contain thousands of documents that are somewhat related to each other but only at a relatively broad level. Fourth, classifications are often static, i.e. articles are classified at the time of publication. If a classification scheme is changed, the papers are not updated or re-classified. Finally, for many domains, expert classifications simply do not exist.

With VCP, the problems could be overcome. (Virtual) citation proximity (1) covers many types of relatedness; (2) allows documents to be in unlimited numbers of co-citation clusters; (3) has no limitations for the number of clusters; (4) is dynamic; and (5) can be learned for any domain that uses citations.

In recent years advances in deep-learning have shown the ability to identify complex patterns in text based data in areas such as translation (Wu et al., 2016) and sentiment analysis (Dos Santos and Gatti, 2014).

A document embedding (Le and Mikolov, 2014; Dai et al., 2015) is an embedding representing an entire document trained using a paragraph embedding model. Document embedding vectors have been shown to be superior to other text representations such as bag-of-words as they take into account the relative positions of the words in the text, although experimental they may be an interesting feature representation to train VCP. Overall, papers with success in using machine learning for dealing with larger passages of text more limited in number (Liu et al., 2018), compared to longer texts (Lopez and Kalita, 2017). Some relevant research was found in the areas of news article recommender systems (Park et al., 2017).

4 Methodology

4.1 VCP Implementation

We implement four VCP variations. The first implementation is a sequential neural network with a CNN and LSTM layer with drop-out. The second, third and fourth implementation are Siamese neural networks, whereas the second implementation consists of two LSTM layers with drop-out (Figure 3); the third implementation consists of a CNN and LSTM layer with drop-out; and the fourth implementation consists of a CNN and LSTM layer with no drop-out. The Siamese architectures finish with a sequential dense layer to join the subnetworks. We choose combinations of 200-neuron

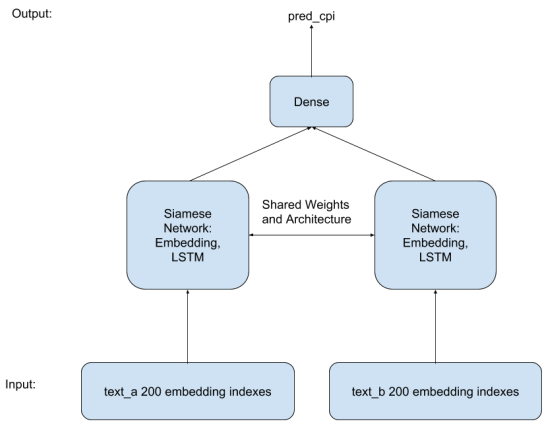


Figure 3: Siamese Neural Network Architecture Diagram.

LSTM and 64-filter CNN layers in both sequential and Siamese architectures.

So far, Siamese networks have been particularly successful in face recognition. During training, the network receives a triplet as input consisting of an anchor image of a person A, another image of the same person, and an image of a person that is not A. The network is trained to learn a similarity or distance function that can express the high similarity (or low distance) of the anchor image and images of the same person, and disimilarity (or high distance) of the anchor image and negative person. Siamese networks also have been successfully used to learn text similarity (Mueller and Thyagarajan, 2016). Siamese architectures facilitate the sister sub-networks to learn high level representations from both input texts first. Then once the Siamese Neural Network has transformed the input into higher level representations they can be combined together again to determine the relationship between the two texts.

In our scenario, triplets consist of an anchor citation and a close co-citation (as both express the same semantic concept) as well as of a document that is dissimilar to the anchor citation. We hypothesize that a neural network that is capable of learning the abstract concept of a "person", based on vastly different images (pixels) of that person, should also be able to learn the abstract semantic concept of relatedness, based on vastly different documents (textual features) and citation proximity.

Each of the four implementations takes as input two documents represented by their title and the first 200 words of the body text, and predicts the distance in which these two documents would be co-cited, if they were co-cited. All VCP variations used the GloVe6B word embedding model to represent textual features. We used Glove6B out-of-box, i.e. trained on a dump from English Wikipedia in 2014, and with 100 dimensions. All four models were implemented in Keras, and trained over 50 epochs. The source code and data is available on GitHub https://github.com/BeelGroup/Virtual-Citation-Proximity/.

We need to emphasize that we did not compare our implementations against a state-of-the-art baseline as there does not exist any other work that predicts citation proximity. Hence, we only compare the performance of our models against a trivial baseline, i.e. the average co-citation proximity in the corpus. In the future, the predicted citation proximity should be used in a recommender system and could then be compared against baselines like content-based filtering .

4.2 Dataset

We initially aimed to use research papers and citations for our experiments. Eventually, we decided to choose Wikipedia as a substitute. Parsing research papers (PDF files) for their in-text citation was too computationally expensive and error prone, and we did not find existing suitable dataset that would have contained enough in-text citation data[3]. Wikipedia contains millions of articles, that are somewhat comparable to research articles, and these articles contain hyperlinks, that are comparable to citations. Also, Wikipedia data is machine readable, i.e. hyperlinks/citations can easily be identified. We used the Wikipedia dump from January 1st 2019 with $\tilde{1}5$ million articles, of which we choose a random sample (filtering out articles co-cited less than 5 times) of 1,000 articles and all articles co-cited with those sample articles. This resulted in 2.1 million co-citation pairs.

A key factor in citation proximity analysis is the question how to exactly measure proximity, or distance. The original authors of Citation Proximity Analysis expressed the distance between two co-citations through a 'citation proximity index' (**CPI**) (Gipp and Beel, 2009). If two documents were co-cited in the same sentence, CPI was 1; if documents were co-cited in the same paragraph, CPI was 0.5; and so on (Table 1). Many more variations have

[3] unarXive (Saier and Färber, 2020) might be suitable, but it was just released after we conducted our experiments

been proposed to calculate CPIs, e.g. (Kim et al., 2016). We follow Schwarzer et al. including their suggested damping factor α of 0.855 to scale word distance (Schwarzer et al., 2016).

$$CPI(a,b) = \sum_{j=1}^{m} \Delta_j(a,b)^{-\alpha},$$
$$\text{with } \Delta_j(a,b)^{-\alpha} = \left\{ \begin{array}{ll} |v_{a,j} - v_{b,j}|^{-\alpha}, & v_{a,j} > 0 \wedge v_{b,j} > 0 \\ 0, & \text{otherwise} \end{array} \right\} \quad (1)$$

Table 1: CPI values for co-cited document pairs, as proposed by the original authors (Gipp and Beel, 2009). However, these values are only for a single occurrence of a co-citation pair. If e.g. documents A and B are co-cited by document C in the same sentence but by document D in different paragraphs, the final CPI value must be a fusion of these CPI values (e.g. the min, max or average).

Occurrence	CPI Value
Sentence	1
Paragraph	1/2
Chapter	1/4
Same journal / same book	1/8
Same journal but different edition	1/16

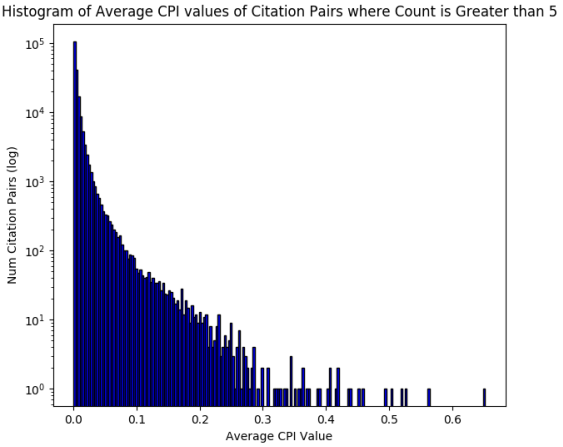


Figure 4: Distribution of CPI Values in the Wikipedia dataset. Many CPI values are very small.

A second important question is how to deal with multiple occurrences of the same co-citation pair in different documents, and hence different CPI values for each occurrence. The most simple solutions are using the minimum, average or sum of the individual CPIs (Knoth and Khadka, 2017). We choose for our work the average CPI as this has been shown to be among the most effective choices typically(Knoth and Khadka, 2017). We calculated CPI values with the tool Citolytics (Schwarzer et al., 2017)[4] as per the equation below, based on Schwarzer et al.. *(a,b)* is a document pair with *m* co-citations and $v_{a,j}$ is the position in words of the *jth* citation of *a*. See example data (Table 2).

4.3 Evaluation Metric

We evaluate the VCP implementations based on how well they predict the actual CPI, which theoretically takes values between 0 and 1, but typically is between 0 and 0.1 (Figure 4). Performance is measured by mean absolute error (**MAE**).

We have not yet conducted additional recommender-system specific experiments. We assume that the more precise the prediction of the CPIs are, the better the recommendation performance becomes. Of course, this is a strong assumption that needs to be validated in future experiments.

5 Results and Discussion

All four models achieved relatively low MAEs between 0.0059 (Sequential 1D CNN + LSTM) and 0.0055 (Siamese LSTM + LSTM; Siamese CNN + LSTM, No Dropout) (Figure 5). All three Siamese Neural Networks outperformed the simple Sequential model CNN+LSTM. The differences among the three Siamese architectures are statistically not significant. All four models performed statistically significant better (p¡0.01; two-tailed t-test) than the baseline, i.e. the mean CPI in the dataset (MAE=0.0069). The low MAEs must be seen with some skepticism. The average of the actual CPI values in the dataset was 0.0069 with data skewed towards smaller values. Hence, an MAE of e.g. 0.0055 is promising (20% lower, i.e. better, than

[4]Citolytics only returns the sum of the individual CPIs, so we calculated average CPIs ourselves

Table 2: Citolytics Wikipedia CPI Pair Dataset Format Example.

Hash	Title A	Title B	Dist	Count	Title A ID	Title B ID	CPI
-124	USA	USSR	312	12	5	7	0.26

Figure 5: Mean Average Error of the four VCP variations and the mean-baseline.

the mean CPI) but not as good as it may seem on first glance.

The learning curves of the four VCP approaches indicates that citation proximity could not be learned very effectively. Figure 6 shows the training and validation error rates of the Siamese CNN + LSTM Model over 50 epochs. The validation error shows that no real learning occurs after the first epoch.

Overall, our result, i.e. a 20% improvement over the trivial 'mean' baseline, is promising but more research is needed to confirm the effectiveness of Virtual Citation Proximity. In the current experiment, we used the average CPI of document pairs as target, but alternatives such as the minimum or maximum CPI might be easier to learn for a Siamese network. Also, there were many documents with low CPI values in the corpus, which might have introduced noise. In future work, we would focus on documents with higher CPI values as we expect their signal to be stronger. We also plan to use more than 200 words in future experiments, as more words might contain more semantic meaning of why a document was cited. Maybe most importantly, Virtual Citation Proximity needs to be evaluated in more recommender-system specific experiments. So far, we 'only' predicted citation distance. The key question, however, is how good

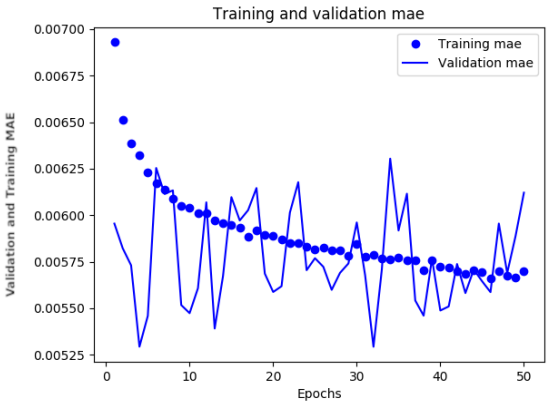


Figure 6: Mean Average Error of Siamese 1D CNN and LSTM over 50 Epochs.

VCP-based recommendations can be, i.e. how precise they need to be to contribute to business value (Jannach and Jugovac, 2019). It will also be interesting to see how VCP compares with content-based filtering, citation-based approaches, and machine learning models trained on expert opinions as ground truth.

While our initial results are 'only' good, we see an enormous potential in Virtual Citation Proximity for improving recommender systems for research papers, web pages, patents, and other document types. We are confident that VCP could become

a new state-of-the-art approach for research paper recommender systems that brings citation-based recommendation effectiveness to the community, applicable to all textual documents. In the best case, VCP might even outperform citation based approaches as VCP learns from both terms *and* citations and hence VCP might be able to learn semantic concepts in a completely new way beyond traditional citation and content analysis.

References

Giovanni Abramo, Ciriaco Andrea DAngelo, and Anastasiia Soldatenkova. 2016. The dispersion of the citation distribution of top scientists publications. *Scientometrics*, 109(3):1711–1724.

A Balaji, S Sendhilkumar, and GS Mahalakshmi. 2017. Finding related research papers using semantic and co-citation proximity analysis. *Journal of Computational and Theoretical Nanoscience*, 14(6):2905–2909.

Joeran Beel. 2017. Virtual citation proximity (vcp): Calculating co-citation-proximity-based document relatedness for uncited documents with machine learning [proposal]. *ResearchGate*.

Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. 2016. Research paper recommender systems: A literature survey. *International Journal on Digital Libraries*, (4):305–338.

Jöran Beel and Bela Gipp. 2010. Link analysis in mind maps: a new approach to determining document relatedness. In *4th International Conference on Uniquitous Information Management and Communication*, page 38. ACM.

Andrew M Dai, Christopher Olah, and Quoc V Le. 2015. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*.

Cicero Dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

Martin G Erikson and Peter Erlandson. 2014. A taxonomy of motives to cite. *Social Studies of Science*, 44(4):625–637.

Michael Färber and Ashwath Sampath. 2019. Determining how citations are used in citation contexts. In *Digital Libraries for Open Knowledge*, pages 380–383, Cham. Springer International Publishing.

Bela Gipp and Jöran Beel. 2009. Citation proximity analysis (cpa): A new approach for identifying related work based on co-citation analysis. In *ISSI09: 12th International Conference on Scientometrics and Informetrics*, pages 571–575.

Bela Gipp, Adriana Taylor, and Jöran Beel. 2010. Link proximity analysis-clustering websites by examining link proximity. In *International Conference on Theory and Practice of Digital Libraries*, pages 449–452. Springer.

Michael Golosovsky. 2017. Power-law citation distributions are not scale-free. *Physical Review E*, 96(3):032306.

Hebatallah A Mohamed Hassan. 2017. Personalized research paper recommendation using deep learning. In *Proceedings of the 25th conference on user modeling, adaptation and personalization*, pages 327–330. ACM.

Dietmar Jannach and Michael Jugovac. 2019. Measuring the business value of recommender systems. *ACM Transactions on Management Information Systems (TMIS)*, 10(4):1–23.

Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender Systems: An Introduction*. Cambridge University Press.

Ha Jin Kim, Yoo Kyung Jeong, and Min Song. 2016. Content-and proximity-based author cocitation analysis using citation sentences. *Journal of Informetrics*, 10(4):954–966.

Petr Knoth and Anita Khadka. 2017. Can we do better than co-citations? In *2nd BIRNDL Workshop, Tokyo, Japan*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Bang Liu, Ting Zhang, Di Niu, Jinghong Lin, Kunfeng Lai, and Yu Xu. 2018. Matching long text documents via graph convolutional networks. *arXiv preprint arXiv:1802.07459*.

Shengbo Liu and Chaomei Chen. 2011. The effects of co-citation proximity on co-citation analysis. In *Proc. of ISSI*, pages 474–484.

Marc Moreno Lopez and Jugal Kalita. 2017. Deep learning applied to nlp. *arXiv preprint arXiv:1703.03091*.

Pasquale Lops, Dietmar Jannach, Cataldo Musto, Toine Bogers, and Marijn Koolen. 2019. Trends in content-based recommendation. *User Modeling and User-Adapted Interaction*, 29(2):239–249.

Shahin Mohammadi, Sudhir Kylasa, Giorgos Kollias, and Ananth Grama. 2016. Context-specific recommendation system for predicting similar pubmed articles. In *16th International Conference on Data Mining*, pages 1007–1014. IEEE.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *30th AAAI Conference on Artificial Intelligence*.

Keunchan Park, Jisoo Lee, and Jaeho Choi. 2017. Deep neural networks for news recommendations. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2255–2258. ACM.

Shengwen Peng, Hiroshi Mamitsuka, and Shanfeng Zhu. 2018. Meshlabeler and deepmesh: Recent progress in large-scale mesh indexing. In *Data Mining for Systems Biology*, pages 203–209. Springer.

Tarek Saier and Michael Färber. 2020. unarxive: a large scholarly data set with publications' full-text, annotated in-text citations, and links to metadata. *Scientometrics*, pages 1–24.

Malte Schwarzer, Corinna Breitinger, Moritz Schubotz, Norman Meuschke, and Bela Gipp. 2017. Citolytics: A link-based recommender system for wikipedia. In *Proceedings of the 11th ACM Conference on Recommender Systems*, pages 360–361.

Malte Schwarzer, Moritz Schubotz, Norman Meuschke, Corinna Breitinger, Volker Markl, and Bela Gipp. 2016. Evaluating link-based recommendations for wikipedia. In *16th ACM/IEEE Joint Conference on Digital Libraries*, pages 191–200.

Peter Willett. 2013. Readers' perceptions of authors' citation behaviour. *Journal of Documentation*, 69(1):145–156.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Citations Beyond Self Citations: Identifying Authors, Affiliations, and Nationalities in Scientific Papers

Yoshitomo Matsubara
University of California, Irvine
yoshitom@uci.edu

Sameer Singh
University of California, Irvine
sameer@uci.edu

Abstract

The question of the utility of the blind peer-review system is fundamental to scientific research. Some studies investigate exactly how "blind" the papers are in the double-blind review system by manually or automatically identifying the true authors, mainly suggesting the number of self-citations in the submitted manuscripts as the primary signal for identity. However, related studies on the automated approaches are limited by the sizes of their datasets and the restricted experimental setup, thus they lack practical insights into the blind review process. Using the large Microsoft Academic Graph, we train models that identify authors, affiliations, and nationalities of the affiliations for anonymous papers, with 40.3%, 47.9% and 86.0% accuracy respectively from the top-10 guesses. Further analysis on the results leads to interesting findings *e.g.*, 93.8% of test papers written by Microsoft are identified with top-10 guesses. The experimental results show, against conventional belief, that the self-citations are no more informative than looking at the common citations, thus suggesting that removing self-citations is not sufficient for authors to maintain their anonymity.

1 Introduction

Scientific publications play an important role in dissemination of advances, and they are often reviewed and accepted by professionals in the domain before publication to maintain quality. In order to avoid unfairness due to identity, affiliation, and nationality biases, peer review systems have been studied extensively (Yankauer, 1991; Blank, 1991; Lee et al., 2013), including analysis of the opinions of venue editors (Brown, 2007; Baggs et al., 2008) and evaluation of review systems (Yankauer, 1991; Tomkins et al., 2017). It is widely believed that a possible solution for avoiding biases is to keep the author identity blind to the reviewers, called double-blind review, as opposed to only hiding the identity of the reviewers, as in single-blind review (Lee et al., 2013). Since some personal information (*e.g.*, author, affiliation and nationality) could implicitly affect the review results (Lee et al., 2013), these procedures are required to keep them anonymous in double-blind review, but this is not foolproof. For example, experienced reviewers could identify some of the authors in a submitted manuscript from the context. In addition, the citation list in the submitted manuscript can be useful in identifying them (Brown, 2007), but is indispensable as it plays an important role in the reviewing process to refer readers to related work and emphasize how the manuscript differs from the cited work.

To investigate blindness in double-blind review systems, Hill and Provost (2003) and Payer et al. (2015) train a classifier to predict the authors, and analyze the results. However, they focus primarily on the utility of self-citations in the submitted manuscripts as a key to identification (Mahoney et al., 1978; Yankauer, 1991; Hill and Provost, 2003; Payer et al., 2015), and do not take author's citation history beyond just self-citations into account. The experiment design in these studies is also limited: they use relatively small datasets, include papers only from a specific domain (*e.g.*, physics (Hill and Provost, 2003), computer science (Payer et al., 2015) or natural language processing (Caragea et al., 2019)), and pre-select the set of papers and authors for evaluation (Payer et al., 2015; Caragea et al., 2019). Furthermore, they focus on author identification, whereas knowing affiliation and the nationality also introduces biases in the reviewing process (Lee et al., 2013).

In this paper, we use the task of author identity, affiliation, and nationality predictions to analyze the extent to which citation patterns matter, evaluate our approach on large-scale datasets in many domains, and provide detailed insights into

the ways in which identity is leaked. We describe the following contributions:

1. We propose approaches to identify the aspects of the citation patterns that enable us to guess the authors, affiliations, and nationalities accurately. To the best of our knowledge, this is the first study to do so. Though related studies mainly suggest authors avoid self-citations for increasing anonymity of submitted papers, we show that overlap between the citations in the paper and the author's previous citations is an incredibly strong signal, even stronger than self-citations in some settings.
2. Our empirical study is performed on (i) a real-world large-scale dataset with various fields of study (computer science, engineering, mathematics, and social science), (ii) study different relations between papers and authors, and (iii) two identification situations: "guess-at-least-one" and "cold start". For the former, we identify authors, affiliations and nationalities of the affiliations with 40.3%, 47.9% and 86.0% accuracy respectively, from the top-10 guesses. For the latter, we focus on papers whose authors are not "*guessable*", and find that the nationalities are still identifiable.
3. We perform further analysis on the results to answer some common questions on blind-review systems: "*Which authors are most identifiable in a paper?*", "*Are prominent affiliations easier to identify?*", and "*Are double-blind reviewed papers more anonymized than single-blind?*". One of the interesting findings is that 93.8% of test papers written by a prominent company can be identified with top-10 guesses.

The dataset used in this work is publicly available, and the complete source code for processing the data and running the experiments is also available.[2]

2 Related work

Here, we summarize related work, and describe their limitations in analyzing anonymity in the blind review systems.

2.1 Citation Analysis and Application

There are several studies that propose applications using citation networks (Dong et al., 2017), and they are not limited to applications of scientific papers in academia. Fu et al. (2015, 2016) study patent citation recommendation and propose a citation network modeling. Levin et al. (2013) introduce new features for citation-network-based similarity metric and feature conjunctions for author disambiguation, and it outperforms the clustering with features from prior work. Fister et al. (2016) define citation cartel as a problem arising in scientific publishing, and they introduce an algorithm to discover the cartels in citation networks using a multi-layer network. Petersen et al. (2010) propose the methods for measuring the citation and productivity of scientists, and examine the cumulative citation statistics of individual authors by leveraging six different journal paper datasets. Though a study of Su et al. (2017) is not a citation related work, it proposes an approach to de-anonymize web browsing histories with social networks and link them to social media profiles. Kang et al. (2018) publish the first dataset of scientific peer reviews, including drafts and the decisions in ACL, CoNLL, NeurIPS and ICLR. Using the published dataset, they also present simple models to predict the accept/reject decisions and numerical scores of review aspects.

2.2 Blind Review and Author Identification

Blind review systems in conferences and journals have been addressed for decades, and have attracted researchers' attention recently (Blank, 1991; Brown, 2007; Lee et al., 2013). For instance, Snodgrass (2006) summarizes previous studies of the various aspects in blind reviewing within a large number of disciplines, and discusses the efficacy of blinding while mentioning how blind submitted/published papers are in different studies. Tomkins et al. (2017) show an example of affiliation bias in the reviewing process. They performed an experiment in the reviewing process of WSDM 2017, which considers the behavior of the program committee (PC) members only, and the members are randomly split into two groups of equal size: single-blind and double-blind PCs. They report that single-blind reviewers bid for 22% more papers, and preferentially bid for papers from top institutions. Bharadhwaj et al. (2020) discuss the relation between de-anonymization of authors through arXiv preprints and acceptance of a research paper at a (nominally) double-blind venue. Specifically, they create a dataset of ICLR 2020 and 2019 submissions, and present key inferences obtained by analyzing the dataset such as "releasing preprints on arXiv has a positive correlation with

[2] https://github.com/yoshitomo-matsubara/guess-blind-entities

acceptance rates of papers by well known authors."

Some studies attempt to manually identify authors and affiliations in submitted manuscripts. Yankauer (1991) sent a short questionnaire the reviewers of American Journal of Public Health for asking them to identify the author and/or institution of submitted manuscripts, and reported that blinding could be considered successful 53% of time. Justice et al. (1998) examine whether masking reviewers to author identity improves the peer review quality. Through a controlled trial for external reviews of manuscripts submitted to five different journals, they conclude that masking fails to the identity of well known authors, and may not improve the fairness of review.

In addition to the manual identification studies, some researchers propose automatic approaches to guess authors in published papers. Table 1 summarizes datasets in other studies. To the best of our knowledge, Hill and Provost (2003) first propose automatic methods using citation information for author identification and perform an experiment with a dataset, that consists of physics papers in the arXiv High Energy Particle Physics between 1992 and 2003. Payer et al. (2015) propose deAnon, a multimodal approach to deanonymize authors of academic papers. They perform experiments with papers in the proceedings of 17 different computer science related conferences from 1996 to 2012. Similarly, Caragea et al. (2019) address a similar research question, and train convolutional neural networks on the datasets of the prefiltered ACL and EMNLP papers, using various types of features such as context, style, and reference.

However, there are some biased observations in their work. As shown in Table 1, one of the biggest concerns lies in their datasets. They use only one major field dataset in their work: physics (Hill and Provost, 2003) , computer science (Payer et al., 2015) and natural language processing (Caragea et al., 2019), but it would be not enough to discuss if their approaches actually work in various fields of study. The second biggest concern is that they understate a possibility that there are also papers where no authors can be found in the training dataset (Payer et al., 2015; Caragea et al., 2019). Especially in Payer et al. (2015)'s work, the authors do not mention the possibility, but achieve 100% accuracy after trying all guesses for each paper in their *guess-one*, *guess-most-productive-one* and *guess-all* scenarios even though it is very difficult in general to find papers where all the authors are seen in the training dataset.

Furthermore, they focus only on productive authors who have at least three papers in the training dataset, and the numbers of candidates in training and test papers can be considered very limited. Similarly, Caragea et al. (2019) exclude any authors with less than three papers from their datasets after an author name normalization process described in Section 4.3. Hill and Provost (2003) argue that there are some test papers for which they did not see the author(s) in their training dataset. However, the lack of true authors' citation histories does not seem to strongly affect their observed matching accuracy, and it can be caused by the scale of the dataset. Also, their studies do not cover either affiliation or nationality (including cold start scenario), which could cause affiliation and nationality biases (Lee et al., 2013) if they are identifiable.

3 Identification Approach

Training and test datasets are independently prepared, and papers in the training dataset are older than those in the test dataset. We extract features from the training dataset to model each author's citation pattern, and the entity also can be affiliation or nationality depending on what we guess in the test papers. Building entity models, we score each entity based on its extracted features for a test paper, and sort the scores for the paper to rank all the entities. We describe the detail of each process in the following sections.

3.1 Citation Features

Scientific papers have references to introduce related work to readers and sometimes compare the results with the work in order to emphasize the difference between them. We assume that authors have their own citation patterns, and it can be a clue to guess authors in papers. They would repeatedly cite the same papers and their own publication if the projects and fields are similar to their previous ones. Also, we assume that the citation list in a paper would not dramatically change between before and after the blind-review process, since we are limited in access to the published papers only.

In addition to citation features (Hill and Provost, 2003), Payer et al. (2015) and Caragea et al. (2019) use contextual features. As discussed in (Narayanan et al., 2012; Rosen-Zvi et al., 2004), author-topic model and writing style would be hints

Table 1: Dataset comparison with other studies.

	Hill and Provost (2003)	Payer et al. (2015)	Caragea et al. (2019)	Our work
Domains	Physics	CS	NLP	All, CS, Eng., Math, Soc. Sci.
#authors	7,424	1,405	262 & 922	22k - 2M
#papers	29,514	3,894	622 & 3,011	231k - 825k

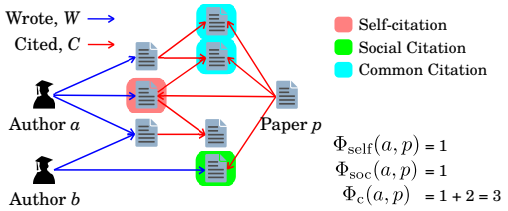


Figure 1: Example of self-, social and common citations $\Phi_{\{\text{self, soc, c}\}}(a, p)$ for author a and paper p.

to identify authors. In this work, however, we only use citation and publication histories for identification. This also reduces computational load in training and test processes and enables us to further analyze the performances in various situations focused on citation features. In the following approaches, the models skip scoring candidate authors (entities) given a test paper if they have no citation features (all zero(s)) since this work focuses on citation pattern in the identification problems.

Figure 1 illustrates an example citation graph with red and blue edges from $x \rightarrow y$ indicating x cited y and x wrote y, respectively. We focus here on three types of citations described in the following sections: self, social, and common citations.

3.2 Self-citations, SC

As discussed in these studies (Mahoney et al., 1978; Yankauer, 1991; Hill and Provost, 2003; Payer et al., 2015), self-citations can be a clue in identification. The Self-citation (SC) model calculates how many papers written by author a are cited by paper p based on his/her publication history

$$\Phi_{\text{self}}(a, p) = \sum_{r \in \text{Ref}_p} W(a, r)\,,$$

$$W(a, p) = \begin{cases} 1 & \text{if } a \text{ wrote } p \\ 0 & \text{otherwise} \end{cases},$$

where p is a blind (test) paper, and a is a candidate author seen in the training dataset. Ref_p is the set of paper IDs cited by paper p. In Figure 1, a wrote three different papers, and one of them is cited by p i.e., $(\Phi_{\text{self}}(a, p) = 1)$, assuming a wrote p.

Hill and Provost (2003) use inverse citation-frequency (icf) for weighted scoring for self-citations to incorporate importance of the self-citation. We include this in our SC model as well:

$$\Phi^{\text{icf}}_{\text{self}}(a, p) = \sum_{r \in \text{Ref}_p} W(a, r) \cdot icf(r) \qquad (1)$$

$$icf(r) = \log\Big(\frac{N_{\text{tr}}}{1 + \sum_{p' \in P_*} C(p', r)}\Big)\,,$$

$$C(p, r) = \begin{cases} 1 & \text{if } p \text{ cited } r \\ 0 & \text{otherwise} \end{cases},$$

where P_* denotes the set of papers in the training dataset, $N_{\text{tr}} = |P_*|$ is the number of papers, and A is the set of all authors in the training dataset.

3.3 Social citations, SocC

Instead of self-citations, it is also common to cite papers written by past collaborators. In this work, we call such citations social citations. Though this model itself will not be as powerful as the SC model, the social citation feature helps us identify potential connections between a test paper and candidates (authors) as this approach covers the publication histories of the past collaborators given an author. Social citation score is defined as:

$$\Phi_{\text{soc}}(a, p) = \sum_{r \in \text{Ref}_p} \sum_{a_c \in A_a} W(a_c, r), \qquad (2)$$

where A_a is the set of authors who wrote a paper with author a. In Figure 1, author a wrote a paper with author b, and p cited a paper written by b. Then, the social citation count is one.

Similar to the SC model, our SocC model uses the weighted score:

$$\Phi^{\text{icf}}_{\text{soc}}(a, p) = \sum_{r \in \text{Ref}_p} \sum_{a_c \in A_a} W(a_c, r) \cdot icf(r). \qquad (3)$$

3.4 Common Citations, CC

Apart from self and social citations, another clue to the identity might be in all past citations (even ones that are not self or social). Common Citation (CC)

Table 2: Features used for our combined model.

Feature Name	Feature Value
Average icf-weighted CC score	$\frac{\Phi_{\text{c}}^{\text{icf}}(a,p)}{\lvert\text{Ref}_p\rvert}$
CC coverage	$\frac{\lvert\text{Ref}_p \wedge \text{Ref}_a^*\rvert}{\lvert\text{Ref}_p\rvert}$
Average SocC score	$\frac{\Phi_{\text{soc}}^{\text{icf}}(a,p)}{\lvert\text{Ref}_p\rvert}$
SocC coverage	$\frac{\lvert\text{Ref}_p \wedge \text{Pub}_{A_a}\rvert}{\lvert\text{Ref}_p\rvert}$
icf-weighted SC score	$\Phi_{\text{self}}^{\text{icf}}(a,p)$
SC score	$\Phi_{\text{self}}(a,p)$

Ref_a^*: set of paper IDs cited by papers written by a in the training dataset, while Pub_{A_a}: set of papers written by past collaborators of author a.

model thus calculates how many times in author a cites each of the papers cited by paper p:

$$\Phi_c(a,p) = \sum_{r \in \text{Ref}_p} \sum_{p'_a \in P_a} C(p'_a, r)\,, \qquad (4)$$

where P_a is the set of a's papers in the training dataset. In Figure 1, the paper p cites two of the papers cited by a, and the author's common citation count is three. We also include a weighted version:

$$\Phi_{\text{c}}^{\text{icf}}(a,p) = \sum_{r \in \text{Ref}_p} \sum_{p'_a \in P_a} C(p'_a, r) \cdot icf(r). \qquad (5)$$

3.5 Learning a Classifier

In addition to separately using the SC, SocC and CC models, we introduce a combined model (Full) that uses all the citation features. We estimate the parameters of features by the mini-batch gradient descent method. Due the cost of computing softmax function over all possible authors for a paper, we use negative sampling, similar to (Mikolov et al., 2013), leading to the following loss:

$$l(\{a_i, p_i\}, \boldsymbol{\theta}) = \frac{1}{K} \sum_{i=1}^{K} \Big(\log \sigma\big(\boldsymbol{\theta} \cdot \boldsymbol{\phi}(a_i, p_i)\big) - \frac{1}{|\bar{A}_{p_i}|} \sum_{\bar{a} \in \bar{A}_{p_i}} \log \sigma\big(\boldsymbol{\theta} \cdot \boldsymbol{\phi}(\bar{a}, p_i)\big) \Big) - \lambda ||\boldsymbol{\theta}||_2^2 \qquad (6)$$

where $\{a_i, p_i\}$ is a set of pairs of authors and their papers, and $\boldsymbol{\theta}$ is 7-dimensional estimated parameter vector. $\boldsymbol{\phi}(a_i, p_i)$ contains a bias term and features shown in Table 2, and K is the batch size. $\bar{A}_{p_i}$ is a set of randomly sampled authors as negative samples given paper p_i, and λ is a hyperparameter for regularization. Note that these parameters $\boldsymbol{\theta}$ are shared across all the authors in the dataset.

4 Experimental Setup

We define some terms and variables used in the following sections, and then describe the MAG dataset and how we develop benchmarks from it.

4.1 Evaluation Setup

We consider three different entity disambiguation scenarios: author, affiliation, and nationality. For each, our primary evaluation metric is *hits at least*, **HAL**M@k, accuracy of our guesses. If our top-k ranking hits at least M of all the true entities in a test paper, it is considered successfully guessed. M is typically fixed at 1 in the related studies (Blank, 1991; Yankauer, 1991; Justice et al., 1998; Hill and Provost, 2003; Payer et al., 2015; Caragea et al., 2019). Similarly, the range of k is 1-100 (Hill and Provost, 2003), 1-1000 (Payer et al., 2015) and 10 (Caragea et al., 2019) in the previous work respectively. We also consider an evaluation where we set k to X, the number of the *true* entities of a test paper (*i.e.*, each test paper has a different X.

Additionally, we differentiate between *guessable* and *not guessable* papers. We call a test paper ***guessable*** if at least M of all the true entities in the training set have any (non-zero) citation feature used in a model. If M is greater than the number of the true entities in a test paper, it is not ***guessable***.

4.2 Dataset: Microsoft Academic Graph

The Microsoft Academic Graph (MAG) is a large heterogeneous graph of academic entities provided by Microsoft. For paper and author entities, Sinha et al. (2015) collect data from publisher feeds (*e.g.*, IEEE and ACM) and web-page indexed by Bing. They also report that often the quality of the feeds from publishers are significantly better, although the majority of their data come from the indexed pages. The MAG was used in the KDD Cup 2016 for measuring the impact of research institutions and in the WSDM Cup 2016 for entity ranking challenge. The MAG is much larger and more diverse than datasets used in related studies (Hill and Provost, 2003; Payer et al., 2015; Caragea et al., 2019), and uses disambiguated entity IDs. Since some authors seem to be assigned to different author IDs though they look identical, we perform author disambiguation in a more conservative method (Section 4.3) than those in the previous work (Hill and Provost, 2003; Caragea et al., 2019). We use the dataset released in February 2016, thus it includes very few papers published in 2016 than in

the years earlier. Some entries do not have all the attributes we need; we discard such entries.

4.3 Author Disambiguation

It would be ideal if an author name uniquely identifies the entity. In practice, however, an author name tends to be directed to different entities, and an entity may correspond to multiple names (*e.g.*, misspelling and shortened names). Hill and Provost (2003) used the dataset[3] released for KDD Cup 2003. Since this dataset does not contain author IDs, they performed author name disambiguation on the dataset by using author's initial of the first name and entire last name, and Caragea et al. (2019) used the same technique.

Though Hill and Provost (2003) consider the method conservative, it seems rather rough when we tried to reproduce the result. We found that there are 12,625 unique author names, and their disambiguation method resulted in 8,625 unique shortened author names. However, 883 of them have potential name conflicts. Taking an example from the result, "Tadaoki Uesugi" and "Tomoko Uesugi" are considered identical as "T Uesugi", but their names look completely different. Another example is with shortened name; there is a conflict between "A Suzuki", "Alfredo Suzuki" and "Akira Suzuki" though it would make sense if there were only one pair of "A Suzuki" and "Alfredo Suzuki" (or "Akira Suzuki") in the dataset.

The MAG dataset contains author IDs, but there still remains some ambiguity of authors. One of the possible reasons is that some authors may have moved to different affiliations and their new author IDs were generated. Leveraging some of the knowledge in KDD Cup 2013 (author disambiguation challenge) (Chin et al., 2013), we merge authors into one entity if and only if they meet all the following conditions: (1) they have identical full names, and (2) have at least one common past collaborator. This policy reduces the number of unique author IDs in our extracted datasets by about 4%. It may be still incomplete, but it is more conservative and would bias our results less than related work (Hill and Provost, 2003; Caragea et al., 2019).

4.4 Extracted Datasets

Since the MAG dataset is significantly larger than the datasets used in the previous studies (Hill and Provost, 2003; Payer et al., 2015; Caragea et al., 2019), we extract five different datasets from the MAG dataset: randomly sampled, computer science, engineering, mathematics, and social science datasets. All these datasets consist of papers published between 2010 and 2016, and we split the datasets into training (from 2010 to 2014) and test (from 2015 to 2016) datasets. As we mentioned in Section 4.2, the original dataset includes few papers published in 2016 due to its release date. Note that the test datasets include over 20% of the test papers all of whose authors are not found in the training datasets since these training and test datasets are independently prepared.

The first dataset (MAG(10%)) is composed of randomly sampled papers to extract 10% of the whole dataset, and it is most diverse with respect to fields of study among the five datasets. All the other datasets are extracted based on the venue list for each field. For efficiency, it is reasonable to filter candidates (and papers in training dataset) by their fields given a paper because reviewers will know the fields of their venues. Here, an extracted candidate has at least one paper published at a venue in the field defined below, and papers in the training dataset consists of papers written by extracted candidates. Though some papers may not be guessable because of the filter, we consider the possibility to keep our experimental design unbiased (*i.e.*, we do not discard test papers responding to the filtered training dataset). For computer science (CS), we extract papers presented at any of the 60 different venues in a list based on CSRankings[4]. We also create lists of conferences based on Scimago Journal & Country Rank[5] for engineering (Eng.), mathematics (Math), and social science (Soc. Sci.), and the lists consist of 60, 60, and 34 venues respectively. Table 3 shows the statistics of each dataset in author identification. Because of few venues of social science in the original dataset, the dataset is smaller than the others, but still larger than those used in the previous studies (Hill and Provost, 2003; Payer et al., 2015; Caragea et al., 2019).

4.5 Entity Conversion

We also use the above datasets for affiliation and nationality identifications (see Tables 4 and 5 for details). Since some papers in the datasets lack affiliation information, we drop papers from the

[3] https://www.cs.cornell.edu/projects/kddcup/datasets.html

[4] http://csrankings.org/

[5] http://www.scimagojr.com/

Table 3: Author Identification: Statistics of training (2010-2014) and test (2015-2016) datasets.

Dataset	Avg. X	# author IDs		# unique papers	
	test	training	test	training	test (guessable)
MAG(10%)	4.97	2,138,060	484,215	715,968	110,565 (34.1%)
CS	3.81	61,621	19,284	449,875	6,363 (64.7%)
Eng.	3.77	45,731	18,537	391,768	6,065 (48.0%)
Math	3.29	29,950	4,957	269,015	1,723 (53.6%)
Soc. Sci.	3.12	22,059	1,737	231,110	603 (28.7%)

Table 4: Affiliation Identification: Statistics of training (2010-2014) and test (2015-2016) datasets.

Dataset	Avg. X	# affiliation IDs		# unique papers	
	test	training	test	training	test (guessable)
MAG(10%)	1.72	12,416	6,441	289,748	34,927 (78.0%)
CS	1.62	8,487	1,506	260,990	5,738 (93.0%)
Eng.	1.50	8,043	1,646	222,229	5,386 (88.6%)
Math	1.51	7,124	698	153,629	1,265 (94.3%)
Soc. Sci.	1.43	6,597	401	128,718	432 (79.8%)

Table 5: Nationality Identification: Statistics of training (2010-2014) and test (2015-2016) datasets.

Dataset	Avg. X	# nationality IDs		# unique papers	
	test	training	test	training	test (guessable)
MAG(10%)	1.16	130	112	190,026	23,579 (75.5%)
CS	1.16	115	64	194,378	4,073 (89.7%)
Eng.	1.17	108	62	168,631	3,738 (83.9%)
Math	1.16	108	49	114,854	895 (91.8%)
Soc. Sci.	1.08	106	34	98,665	322 (73.6%)

training and test datasets used in affiliation identification if we cannot find at least one affiliation in each of the papers. Since the original dataset does not have nationality information for each affiliation, we perform substring matching for affiliation name based on the information by LinkedIn[6] and Webometrics[7] in order to convert an affiliation to its nationality. Similarly, we drop papers from nationality identification if we cannot find at least one nationality in each of the papers. Note that industrial affiliations may have their offices at several countries, and therefore it is difficult to use their names when converting an affiliation to its nationality. For this reason, we use academic affiliations only in affiliation identification.

Basically, each reference paper can be cited by several published papers, and similarly each published paper can be written by several authors. In contrast, each author (ID) belongs to an affiliation (ID), and an academic affiliation is in a nationality. For this dataset, we can also say that the nationality-affiliation and affiliation-author relationships are single-to-single, and the author-published paper and published paper-reference paper relationships are single-to-many. Authorship and citations of an affiliation are the total papers/citations of their authors, respectively, and similarly for authorship/citations of a nationality.

[6] https://www.linkedin.com/
[7] http://www.webometrics.info/

4.6 Baseline approaches

We extract several sub-datasets based on fields of study from the original dataset. Since the scale of the dataset depends on the field, we use a random scoring approach (Rand) as a baseline to relatively evaluate performance for each dataset. The score is randomly generated between 0 and 1. We also use another random scoring approach (Rand(S)) that skips scoring the candidate authors in a test paper if their citation histories do not include any of the papers cited by the test paper. Since the SC model is based on Hill and Provost (2003), it is also a baseline approach.

5 Experiments and Results

Using various approaches explained above, we perform experiments in two different identification scenarios: "guess-at-least-on" and "cold start". Through the first experiment, we show how anonymized a paper is in each of author, affiliation and nationality identifications. In the second experiment, we show that there remain identity leaks even when no authors in a paper are identifiable.

5.1 *Guess-At-Least-One* Identification

In this experiment, we aim to guess at least one author / affiliation / nationality ($M = 1$), and evaluate HAL1 performances of the five different approaches. If our top k ranking (guesses) includes at least one author in a given paper, the guess is considered successful. Obviously, a paper is less anonymous if we can identify at least one entity (author / affiliation / nationality) in the paper with few guesses. Tables 6-10 show identification performances with five different datasets. The average of Xs and the percentage of the guessable papers in each dataset are given in Tables 3-5.

Overall, our combined model consistently achieves the best performances in the author identification with the datasets, and in the affiliation and nationality identifications the performances of the common citation approach are comparable to those of our combined model. As for the social citation approach, interestingly, it performs better in author identification than in affiliation and nationality identifications though all the other approaches

Table 6: Guess-At-Least-One Scenario: Identification performances with randomly sampled dataset.

MAG(10%)	**Author Identification [%]**				**Affiliation Identification [%]**				**Nationality Identification [%]**		
Top	X	10	100	1000	X	10	100	1000	X	10	100
Rand	0.003	0.0009	0.01	0.089	0.028	0.123	1.37	12.7	1.10	8.60	79.7
Rand(S)	1.63	2.67	12.5	27.8	2.66	9.20	31.2	42.8	11.3	52.8	75.5
SC	8.33	9.71	10.8	10.8	5.67	7.25	7.34	7.34	11.1	12.9	12.9
SocC	6.95	8.62	11.3	11.7	0.544	1.60	7.76	18.7	0.674	3.72	16.5
CC	12.4	15.4	25.5	31.7	11.5	22.9	38.6	42.9	37.3	71.1	75.5
Full	**13.4**	**16.5**	**26.8**	**32.9**	**12.0**	**23.6**	**40.1**	**48.8**	**37.6**	**71.7**	**77.9**

Table 7: Guess-At-Least-One Scenario: Identification performances with computer science dataset.

CS	**Author Identification [%]**				**Affiliation Identification [%]**				**Nationality Identification [%]**		
Top	X	10	100	1000	X	10	100	1000	X	10	100
Rand	0.00	0.015	0.283	2.81	0.00	0.157	2.04	18.1	1.74	10.5	88.8
Rand(S)	2.40	5.30	25.7	55.1	2.09	9.22	46.2	74.2	9.18	51.1	89.7
SC	27.0	34.2	38.1	38.1	**23.4**	37.4	38.3	38.3	45.1	56.6	56.6
SocC	15.5	23.3	38.6	43.5	1.17	4.98	30.1	68.3	1.17	6.41	59.4
CC	24.6	33.9	52.3	60.5	20.1	43.7	69.2	74.2	**54.1**	85.1	89.7
Full	**30.3**	**40.3**	**56.4**	**63.9**	22.7	**47.9**	**71.3**	**79.9**	43.1	**86.0**	**93.0**

Table 8: Guess-At-Least-One Scenario: Identification performances with engineering dataset.

Eng.	**Author Identification [%]**				**Affiliation Identification [%]**				**Nationality Identification [%]**		
Top	X	10	100	1000	X	10	100	1000	X	10	100
Rand	0.00	0.033	0.313	3.13	0.037	0.149	2.01	17.7	1.39	11.7	93.5
Rand(S)	3.15	6.43	25.6	44.3	2.64	10.4	42.8	58.9	10.2	53.6	83.9
SC	19.0	22.1	22.9	22.9	15.0	21.1	21.2	21.2	31.4	37.1	37.1
SocC	9.73	14.5	22.4	23.5	0.613	2.73	17.8	45.7	0.00	1.55	25.2
CC	18.9	25.3	39.5	44.9	15.4	32.1	53.8	58.9	**44.4**	78.3	83.9
Full	**22.4**	**29.8**	**42.4**	**47.7**	**16.7**	**34.6**	**56.2**	**66.4**	40.5	**79.5**	**88.6**

Table 9: Guess-At-Least-One Scenario: Identification performances with mathematics dataset.

Math	**Author Identification [%]**				**Affiliation Identification [%]**				**Nationality Identification [%]**		
Top	X	10	100	1000	X	10	100	1000	X	10	100
Rand	0.00	0.058	0.464	3.31	0.00	0.158	2.37	21.0	1.34	9.60	94.0
Rand(S)	3.83	7.66	31.5	51.2	2.92	13.2	51.9	70.4	10.8	59.7	91.8
SC	23.7	27.0	27.3	27.3	21.6	28.2	28.3	28.3	35.9	43.6	43.6
SocC	11.7	19.4	26.5	27.3	0.395	2.92	19.7	48.3	0.670	5.47	42.7
CC	22.1	32.8	46.5	51.2	20.6	43.1	67.2	70.4	**50.7**	87.0	91.8
Full	**26.5**	**36.3**	**49.4**	**53.6**	**22.5**	**46.0**	**69.7**	**78.7**	47.6	**87.6**	**94.3**

Table 10: Guess-At-Least-One Scenario: Identification performances with social science dataset.

Soc. Sci.	**Author Identification [%]**				**Affiliation Identification [%]**				**Nationality Identification [%]**		
Top	X	10	100	1000	X	10	100	1000	X	10	100
Rand	0.00	0.00	0.166	2.32	0.00	0.00	2.78	19.2	2.17	9.94	97.2
Rand(S)	3.65	6.14	18.2	26.5	3.94	8.30	27.8	38.2	15.8	53.7	73.6
SC	14.1	16.6	17.1	17.1	**14.8**	19.4	19.4	19.4	34.8	36.6	36.6
SocC	7.13	9.95	15.4	15.8	1.85	4.40	13.7	32.9	0.00	1.55	25.2
CC	12.8	17.9	24.2	26.9	11.1	24.8	35.9	38.2	**51.2**	69.9	73.6
Full	**15.4**	**21.1**	**26.7**	**28.7**	13.4	**27.8**	**39.4**	**46.5**	**51.2**	**71.1**	**79.8**

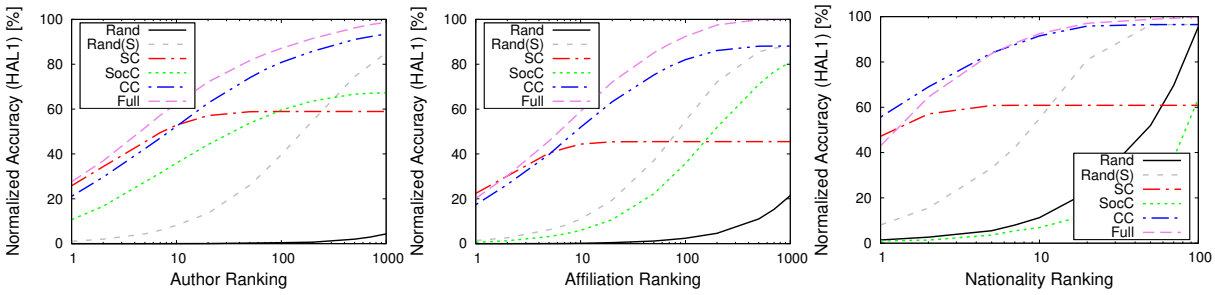


Figure 2: **Author (a), Affiliation (b) and Nationality (c) Identifications:** Normalized performances (divided by the percentage of guessable papers = 64.7, 84.3, 89.7[%] respectively) of five different approaches with CS dataset.

perform best in nationality identification. In addition, as we expected, filtering training datasets (candidates) by venues (fields of study) is effective to guess blind entities in papers of the fields though it is more difficult to guess entities in papers of the randomly sampled and social science datasets because of their smaller percentages of the guessable papers in the datasets.

Figure 2 illustrate the relations between rankings and normalized accuracies with the computer science dataset in author, affiliation and nationality identifications. The self-citation performances converge faster than other approaches using common citation, and this implies that test papers are more likely to have common citations than self-citations. In addition, the performance difference between the SC and our CC (and combined) models are significantly increasing after top 10 choices. Compared to author and affiliation identifications, the number of candidate countries in nationality identification is much smaller, and it could help us easily guess nationalities in test papers.

Some previous studies (Mahoney et al., 1978; Yankauer, 1991; Hill and Provost, 2003; Payer et al., 2015) argue that citing their own papers can be a clue to guess them in their submitted manuscript, and Hill and Provost (2003) reported that their self-citation based method outperforms their common citation based method in the experiment (the *Guess-At-Least-One* scenario). As shown in Tables 6-10, however, there are few significant differences between the accuracy with top 10 or fewer guesses by the CC and SC approaches in author identification. Furthermore, the CC approach outperforms the SC approach in affiliation (with top 10 or more guesses) and nationality (with top X or more guesses) identifications. From these results, it is confirmed that not only self-citation but also common citation can be a clue to identify blind entities in a paper. In other words, we need to decrease both of the numbers of self-citations and common citations if we want to increase anonymity of our submitted manuscripts in the blind review process.

Table 11: Cold Start: Identification for top 10 guesses.

Top-10	**Affiliation** [%]				**Nationality** [%]			
	SC	SocC	CC	Full	SC	SocC	CC	Full
MAG(10%)	1.19	0.715	9.42	**9.59**	6.28	3.27	61.8	**62.2**
CS	7.57	2.66	13.9	**15.4**	25.1	5.62	65.8	**66.5**
Eng.	4.18	1.32	9.68	**10.2**	17.1	6.93	62.8	**63.4**
Math	7.03	1.90	16.2	**16.9**	22.8	6.52	76.1	**76.9**
Soc. Sci.	4.78	1.36	6.83	**7.51**	22.8	2.34	59.8	**60.3**

5.2 Identification in Cold Start Scenario

In the previous author identification problem, we can see from Table 3 that approximately 35-70% of test papers in the datasets are not *guessable* as they do not have any link to at least one of the true authors in the training datasets. The affiliations and nationalities in such test papers, however, may be still guessable since other authors who belong to the affiliation and/or other affiliations in the same country may have similar citation history. In this section, we focus on non-*guessable* test papers in the author identification experiment, and guess the true affiliations and nationalities.

In affiliation identification with non-*guessable* papers for author identification, we ignore papers all of whose authors' affiliations are missing in the datasets, and similarly ignore papers in nationality identification all of whose affiliations could not be converted to their counties. As for training, we use the same training datasets and parameters used in Section 5.1. Table 11 shows the performances of our approaches with top 10 guesses and the percentages of guessable papers in affiliation and nationality identifications. The performances of affiliation and nationality identifications in the cold start scenario for author identification are worse than those

Figure 3: Relation between identification rates (top 10 guesses) and author sequence numbers with CS dataset.

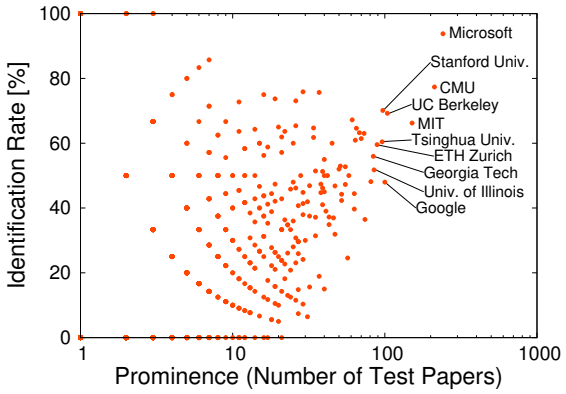


Figure 4: Affiliation prominences and identification rates (top 10 guesses) with CS dataset.

in Tables 6-10. However, at least nationality is still identifiable with a small number of guesses in all the datasets even when we cannot guess true authors in a test paper. Furthermore, we find that the self-citation (SC) model is not useful in this scenario even compared to another baseline approach Rand(S) in nationality identification.

6 Further Analysis

In Section 5, all the entity types are identifiable with a small number of guesses. However, we provide further analysis of the combined model on the CS dataset to answer the following questions.

Which authors are most identifiable?

Figure 3 shows identification rates of different author positions for test papers that have at most 5 authors (85% of the test dataset). As shown, the last author in a paper consistently turns out to be most identifiable, and this may be because the last author is likely to be a director of the research group who may have a stronger research background.

Are prominent affiliations easier to identify?

Here, we consider the number of test papers written by researchers in an affiliation as its prominence. It is apparent from Figure 4 that identification rates of prominent affiliations tend to be high. For example, 93.8% and 77.5% of test papers written by Microsoft and Carnegie Mellon University respectively are identified with top 10 guesses. Note that there are 1,506 affiliations in the graph, but most of the points are overlapped each other.

Are double-blind reviewed papers more anonymized than single-blind reviewed ones?

As shown in Table 12, the performances for papers at single- and double-blind review conferences are almost the same as author and affiliation identifications. This similar performance suggests that the level of anonymity in venues with single-blind review is comparable to that with double-blind review. We only use conferences with at least 40 test papers for denoising here, however, they account for 95% of all test papers.

Table 12: Average percentages of identified papers (top 10 guesses) for single- and double-blind review venues.

CS	**Macro average** [%]		**Micro average** [%]	
Blind review	Single	Double	Single	Double
Author	43.3	42.9	38.3	40.9
Affiliation	55.0	51.9	46.1	48.1

7 Conclusions

The blind review systems are fundamental for research communities to maintain the quality of the published studies. However, it is unclear to what extent the submissions maintain anonymity and how fair the review processes are. In this work, we focus on one of the aspects of de-anonymization by investigating the extent to which we can predict author identity from the paper's citations. Through practical large-scale experiments, we show we can identify author identity, affiliation, and nationality with a few guesses. These results indicate that merely omitting author names is not a sufficient guarantee of anonymity, and may not alleviate fairness considerations in blind review process. This study only involves published papers; analyzing submissions for double-blind review requires considerable involvement of the research communities since they are not public (Tomkins et al., 2017).

Acknowledgements

We thank the anonymous reviewers for their comments. This work is supported in part by a grant from the National Science Foundation (NSF) #IIS-1817183 and #CCRI-1925741. The views in this work do not reflect those of the funding agencies.

References

Judith Gedney Baggs, Marion E. Broome, Molly C. Dougherty, Margaret C. Freda, and Margaret H. Kearney. 2008. Blinding in peer review: The preferences of reviewers for nursing journals. *Journal of Advanced Nursing*, 64(2):131–138.

Homanga Bharadhwaj, Dylan Turpin, Animesh Garg, and Ashton Anderson. 2020. De-anonymization of authors through arxiv submissions during double-blind review.

Rebecca M. Blank. 1991. The Effects of Double-Blind versus Single-Blind Reviewing: Experimental Evidence from The American Economic Review. *The American Economic Review*, 81(5):1041–1067.

Richard J C Brown. 2007. Double anonymity in peer review within the chemistry periodicals community. *Learned Publishing*, 20(2):131–137.

Cornelia Caragea, Ana Uban, and Liviu P. Dinu. 2019. The myth of double-blind review revisited: ACL vs. EMNLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2317–2327. Association for Computational Linguistics.

Wei-Sheng Chin, Yu-Chin Juan, Yong Zhuang, Felix Wu, Hsiao-Yu Tung, Tong Yu, Jui-Pin Wang, Cheng-Xia Chang, Chun-Pai Yang, Wei-Cheng Chang, Kuan-Hao Huang, Tzu-Ming Kuo, Shan-Wei Lin, Young-San Lin, Yu-Chen Lu, Yu-Chuan Su, Cheng-Kuang Wei, Tu-Chun Yin, Chun-Liang Li, Ting-Wei Lin, Cheng-Hao Tsai, Shou-De Lin, Hsuan-Tien Lin, and Chih-Jen Lin. 2013. Effective String Processing and Matching for Author Disambiguation. In *Proceedings of the 2013 KDD Cup 2013 Workshop*, pages 7:1–7:9.

Yuxiao Dong, Hao Ma, Zhihong Shen, and Kuansan Wang. 2017. A Century of Science: Globalization of Scientific Collaborations, Citations, and Innovations. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1437–1446.

Iztok Fister, Iztok Fister, and Matjaž Perc. 2016. Toward the Discovery of Citation Cartels in Citation Networks. *Frontiers in Physics*, 4(December):1–5.

Tao-Yang Fu, Zhen Lei, and Wang Chien Lee. 2015. Patent Citation Recommendation for Examiners. In *Proceedings of the 15th IEEE International Conference on Data Mining*, pages 751–756.

Tao-Yang Fu, Zhen Lei, and Wang-chien Lee. 2016. Modeling Time Lags in Citation Networks. In *Proceedings of the 16th IEEE International Conference on Data Mining*, pages 865–870.

Shawndra Hill and Foster Provost. 2003. The Myth of the Double-Blind Review? Author Identification Using Only Citations. *ACM SIGKDD Explorations Newsletter*, 5(2):179–184.

Amy C Justice, Mildred K Cho, Margaret A Winker, and Jesse A Berlin. 1998. Does Masking Author Identity Improve Peer Review Quality? A randomized controlled trial. *JAMA*, 280(3):240–242.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Carole J. Lee, Cassidy R. Sugimoto, Zhang Guo, and Blaise Cronin. 2013. Bias in Peer Review. *Journal of the American Society for Information Science and Technology*, 14(4):90–103.

Michael Levin, Stefan Krawczyk, Steven Bethard, and Dan Jurafsky. 2013. Citation-Based Bootstrapping for Large-Scale Author Disambiguation. *Journal of the American Society for Information Science and Technology*, 14(4):90–103.

Michael J. Mahoney, Alan E. Kazdin, and Martin Kenigsberg. 1978. Getting Published. *Cognitive Therapy and Research*, 2(1):69–70.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2, pages 3111–3119.

Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. 2012. On the Feasibility of Internet-Scale Author Identification. In *Proceedings of the 33th IEEE Symposium on Security and Privacy*, pages 300–314.

Mathias Payer, Ling Huang, Neil Zhenqiang Gong, Kevin Borgolte, and Mario Frank. 2015. What You Submit Is Who You Are: A Multimodal Approach for Deanonymizing Scientific Publications. *IEEE Transactions on Information Forensics and Security*, 10(1):200–212.

Alexander M. Petersen, Fengzhong Wang, and H. Eugene Stanley. 2010. Methods for measuring the citations and productivity of scientists across time and discipline. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 81(3):1–9.

M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. 2004. The Author-Topic Model for Authors and Documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 487–494.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, , and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In *Proceedings of the 24th International Conference on World Wide Web*, pages 243–246.

Richard Snodgrass. 2006. Single- Versus Double-Blind Reviewing: An Analysis of the Literature. *ACM SIGMOD Record*, 35(3):8–21.

Jessica Su, Ansh Shukla, Sharad Goel, and Arvind Narayanan. 2017. De-anonymizing Web Browsing Data with Social Networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1261–1269.

Andrew Tomkins, Min Zhang, and William D. Heavlin. 2017. Reviewer bias in single- versus double-blind peer review. *Proceedings of the National Academy of Sciences*, 114(48):12708–12713.

Alfred Yankauer. 1991. How Blind Is Blind Review? *American Journal of Public Health*, 81(7):843–845.

SmartCiteCon: Implicit Citation Context Extraction from Academic Literature Using Supervised Learning

Chenrui Guo
Haoran Cui
Wuhan University
Wuhan, Hubei, China
chenruiguo@whu.edu.cn
haoran.cui@whu.edu.cn

Li Zhang
Jiamin Wang
Wei Lu
Wuhan University
Wuhan, Hubei, China
weilu@whu.edu.cn

Jian Wu
Old Dominion University
Norfolk, VA, USA
jwu@cs.odu.edu

Abstract

We introduce SmartCiteCon (SCC), a Java API for extracting both explicit and implicit citation context from academic literature in English. The tool is built on a Support Vector Machine (SVM) model trained on a set of 7,058 manually annotated citation context sentences, curated from 34,000 papers in the ACL Anthology. The model with 19 features achieves F_1=85.6%. SCC supports PDF, XML, and JSON files out-of-box, provided that they are conformed to certain schemas. The API supports single document processing and batch processing in parallel. It takes about 12–45 seconds on average depending on the format to process a document on a dedicated server with 6 multithreaded cores. Using SCC, we extracted 11.8 million citation context sentences from ∼33.3k PMC papers in the CORD-19 dataset, released on June 13, 2020. The source code is released at https://gitee.com/irlab/SmartCiteCon.

1 Introduction

Citations are ubiquitous in scientific publications. With proper citations, statements in research papers are supported by existing works, and readers obtain relevant information beyond the current paper. Citations also form graphs, which provide unique models for ranking, sentimental classification, and plagiarism detection. Therefore, citation analysis plays an important role in helping to understand the deep connection between literature. Accurate citation context recognition is the prerequisite of many downstream applications. Recently, citation context, the text segment that appears around the citation mark in the body text, has been used for enhancing and improving keyphrase extraction (Caragea et al., 2014) and document summarization (Cohan and Goharian, 2015).

There are two types of citation context. Explicit citation contexts (ECC) are sentences containing citation marks. Each citation thus corresponds to one explicit citation context sentence. Implicit citation contexts (ICC) are sentences that are semantically relevant to the cited articles but do not contain citation marks. ICC may appear before or after but may not immediately precede or follow the ECC sentence. One paper could be cited multiple times and each time may have different citation contexts. In the example below, the ECC, containing the citation mark "(Ma et al. 2004)", is highlighted in green. The ICC sentences are highlighted in yellow. The nonhighlighted sentence is not a citation context for the given citation.

> We investigate the impact of semantic constraints on statistical word alignment models as prior knowledge. In (Ma et al. 2004), bilingual semantic maps are constructed to guide word alignment. The framework we proposed seamlessly integrates derived semantic similarities into a statistical word alignment model. And we extended monolingual latent semantic analysis in bilingual application.

Most existing tools extract ECC, i.e., sentences containing citation marks. Although the results are highly relevant, the method omits ICC if the author uses multiple sentences to summarize the results. To our best knowledge, there are no off-the-shelf tools dedicated to ICC extraction. Unlike ECC sentences with citation marks, the lack of explicit marks makes citation context recognition challenging.

In this work, we develop a Java API that implements a supervised machine learning model trained on 7058 manually labeled sentences to extract both ECC and ICC. The model achieves an F_1-measure of 85.6%. The Java API can be deployed on a local machine or as a web service.

2 Related Work

Several citation context extraction methods have been developed. In Nanba and Okumura (1999), the scope of the citation context covered several consecutive sentences before and after the sentences with citation marks (i.e., citation sentence), identified based on a referential relationship with the citation sentence. In another work, Markov model was used for identifying citation context (Qazvinian and Radev, 2010). Sugiyama (2010) described a support vector machine (SVM) and maximum entropy (ME) model for identifying citation sentences using shallow features such as proper nouns and contextual classification of the previous and next sentence (Sugiyama et al., 2010). They found that the performances of SVM and ME do not exhibit significant differences. The positive samples were selected as sentences including citation marks using regular expression matching, ICC extraction was not covered.

ParsCit is an open-source software commonly used for citation parsing and citation context extraction (Councill et al., 2008). ParsCit parses citation strings using a Conditional Random Field (CRF) model. The citation context extraction was performed by extracting a fixed window size of 200 characters on either side of the citation mark. GROBID (Lopez, 2009) is a library to extract information from scholarly documents. The documentation reports the F_1-measure of citation context resolution is around 75%, which counts both the correct identification of citation marks and its correct association with bibliographic references.

In summary, existing citation context extraction tools focus on ECC but ignore ICC, the latter of which includes more sentences semantically related to the cited papers.

3 Supervised Machine Learning Model

Our system is based on a supervised machine learning model proposed in Lei et al. (2016), which classifies a sentence into ICC and non-ICC.

We adopted the ground truth built by Lei et al. (2016) containing 130 articles from 34,000 computational linguistics conference proceedings in ACL Anthology. The original PDF files were converted to XML format using OCR (Schäfer and Weitz, 2012). The training set was labeled by 13 graduate students majoring in information management. The labeling agreement was tested using Cohen's Kappa Coefficient ($\kappa = 0.937$). The final ground truth contains 3,578 positive and 3,480 negative samples. The preprocessing uses Apache OpenNLP for sentence segmentation. Citation marks are identified using regular expressions. Citation marks are then removed, and the original sentences are converted into regular sentences for following analyses such as part-of-speech (POS) tagging. Each sentence is represented by up to 19 features of four types (Table 1). The best model using all features achieves 86% F_1-measure in a the 10-fold cross validation. The SVM outperformed CRF by about 5% in F_1-measure (Table 2).

4 Architecture

The SCC system completes the extraction in four steps (Figure 1): (1) file type recognition, (2) preprocessing, (3) feature extraction, and (4) sentence classification. The output is a JSON file containing ECC and ICC and other citation-related information. The API was written based on the Springboot framework in Java. The machine learning model was implemented with WEKA.

4.1 File Type Recognition

SCC first recognizes the uploaded file type. For a PDF file, SCC invokes GROBID and converts it to an XML file under the TEI schema. If an XML file is uploaded as input, SCC checks whether the schema is in compliance with TEI or PloS ONE schema and passes it to corresponding preprocessors. If a JSON file is uploaded, it checks if it is in compliance with the S2ORC schema, published by Semantic Scholar (Lo et al., 2020). We apply Apache Tika to identify file format. Other format of data files will not be processed.

4.2 Preprocessing

The preprocessing step reads files passed from the last step with customized preprocessors depending on the schema and prepares a canonicalized XML for feature extraction. This step includes the following modules.

4.2.1 Tag removal

This module involves removing irrelevant tags from the DOM structure in the XML file. For example, in the PloS ONE XML files, the `<fig>`, `<sub>`, and `<italic>` tags used for marking up figures, superscripts, and italic font are all moved. Only the text inside these tags are retained. The `<xref>` tags mark positions of citations, which will be used for

#	Features	Categories
1	Distance to the citation sentence	Location
2	In the same paragraph as the citation sentence	Location
3	Include any citation marks	Location
4	The preceding sentence is not a citation sentence	Location
5	The following sentence is not a citation sentence	Location
6	The preceding sentence is the first in paragraph	Location
7	Is the first sentence in the paragraph	Location
8	Section the sentence is in	Location
9	Is the last sentence in the paragraph	Location
10	Trigram Jaccard similarity	Content
11	Bigram Jaccard similarity	Content
12	Unigram Jaccard similarity	Content
13	Include author names	Reference
14	Include any words in the citation sentence	Reference
15	Include He/She/It or their variants	Reference
16	Include Lexical hooks (Murray, 2015)	Reference
17	Include Work Nouns (Murray, 2015)	Reference
18	Number of citation marks	Type
19	Include certain conjunction	Structure

Table 1: Features of the SVM model. A citation sentence is the sentence containing a citation mark.

Model	Precision	Recall	F1-measure
SVM_19	85.6%	85.6%	85.6%
CRF_19	82.2%	79.9%	80.8%

Table 2: Evaluation of SVM and CRF models on 19 features.

restoring citations. We use a separate data structure to store the positions of `<xref>` tags before removing them.

4.2.2 Sentence segmentation

We compared five commonly used sentence segmentation tools, including the Pragmatic Segmenter by Kevin Dias[1], lingpine[2], NLTK[3], a regular expression parser, and the Stanford CoreNLP (Manning et al., 2014) sentence splitter. The golden standard contains 52 sentences provided by Kevin Dias, which covers most possible sentence forms. According to Dias' comparison, the Pragmatic Segmenter receives an accuracy of 98% and the Stanford CoreNLP's accuracy is 59.6%. In our experiments, the accuracies for Lingpipe, NLTK, and regular expression parsers are 61.5%, 50.0%, and 38.5%, respectively. The Pragmatic Segmenter is implemented by Ruby on Rails. To make our API less dependent on a second programming language, we decided to employ Lingpipe for sentence segmentation. We select up to five sentences before and after the current citation sentence as the candidates for classification. This covers almost all sentences that could be classified as ICC.

4.2.3 Canonicalization

Because the input XML may have different schemas, this module takes the processed documents from the above modules and transforms them into a unified schema for feature extraction. The canonicalized schema defines new IDs for chapters, paragraphs, sentences, and citations. The canonicalized XML also includes whether the current sentence contains citation marks.

4.3 Feature Extraction and Text Classification

This step extracts 19 features (Table 1) from the canonicalized XML files and represents each candidate sentence as a vector saved in Livsvm files[4].

[1] https://github.com/diasks2/pragmatic_segmenter

[2] http://www.alias-i.com/lingpipe/

[3] https://www.nltk.org/

[4] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

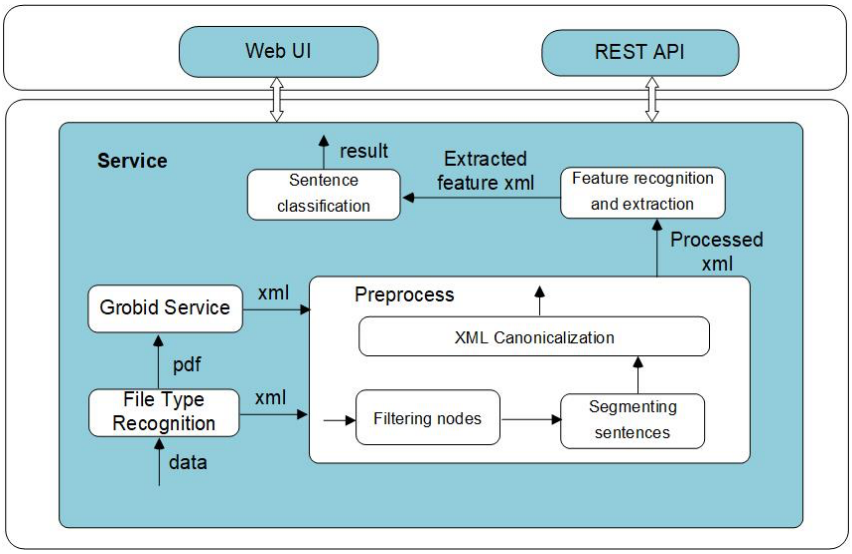


Figure 1: SmartCiteCon architecture.

The SVM model classifies each sentence and outputs a binary indicating whether a sentence is ICC or not. The output JSON file contains citation marks and their positions, citation sentences, and sentences classified as ICC.

4.3.1 User Interfaces

Users can install SCC on a local machine. The API interface supports 3 modes:

1. Single document mode – using the `/extract` service;
2. Batch extraction model with files zipped and transferred through TCP/IP – using `/batchExtract`;
3. Local extraction model with files retrieved from a local directory – using `/localExtract`.

In the single document and batch extraction modes, the API will return JSON objects and execution status. For the local extraction mode, the API will return the execution status and the results will be saved in JSON files.

5 SCC API Performance

We test the SCC API on a computer with 16GB RAM and an Intel Core i7-8570H CPU@2.20GHz, which has 6 hyperthreaded cores (12 threads in total). In a preliminary experiment, we compare the runtime of processing 10 XML documents using a single process under different JVM heap sizes. The runtimes corresponding to 12GB, 8GB, 4GB, and 2GB are 23.8 min, 12.7 min, 7.3 min, and 7.6 min, respectively. Higher heap does not boost processing speed probably due to garbage collection. Based on the results, in the following experiments, 4GB heap was allocated to JVM. The experiments were set to extract citation context from randomly selected documents in different formats. The datasets include 10 PDF documents from PLoS ONE, 10 XML documents corresponding to the PDF documents, and 10 JSON documents from the CORD-19 dataset. We monitor the system using Jprofiler (version 11) and calculate the median time it takes for processing one document as we vary the number of processes N_p. Figure 2 shows that the CPU utilization increases from about 10% and saturates when N_p reaches 8. The memory utilization climbs up slowly as N_p increases but are mostly well below the maximum allocated heap, because processed documents are not stored in memory anymore. The average processing time for all three types gradually decreases as N_p increases but in general, it takes longer to process PDF files than JSON and XML files. The maximum and minimum processing time are shown in Table 3. The runtime can be further reduced by running the API on a computer with more processes on a multicore server. On average, JSON files take the least time to process.

6 Extracting Citation Context from CORD-19

SCC is different from similar tools such as ParsCit and GROBID in that it extracts both ECC and ICC.

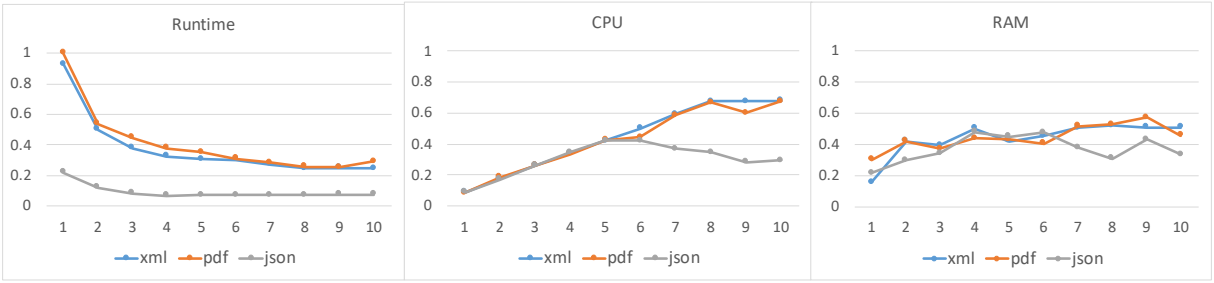


Figure 2: The performance of SCC on a multicore computer. Runtime is normalized at the 177 seconds; the middle panel shows the CPU utilization monitored by Jprofiler; the right panel shows the memory utilization normalized at 4GB.

XML		PDF		JSON	
Max	Min	Max	Min	Max	Min
164	43	177	45	39	12

Table 3: Runtime in seconds for different document formats. The maximum and the minimum runtime are achieved at $N_p = 1$ and $N_p = 8$, respectively.

We apply SCC and extract ECC and ICC from the **CORD-19** dataset. CORD-19 is an open-access dataset compiled by Allen Institute of Artificial Intelligence about COVID-19, SARS, MERS, and related keyphrases conforming to the S2ORC schema (Lo et al., 2020). We downloaded the data released on June 13, 2020 including 50,818 and 69,646 full text papers under the PMC and the PDF folders respectively. The PMC folder contains full-text files obtained by parsing JATS[5] XML files available for PMC papers using a custom parser, generated to the same target output JSON format. This resulted in 1,605,695 ECC and 10,215,848 ICC sentences from 33,319 documents. A fraction of documents was not processed due to the lack of citation marks and runtime exceptions.

SCC code is released at `https://gitee.com/irlab/SmartCiteCon`. The dataset is available on Microsoft OneDrive with a link on the code repository.

7 Lessons Learned

The results in Table 3 indicate that SCC takes about 45 seconds on average to process a PDF document, which is still relatively slow. Using Jprofiler, we found that more than 90% time was spent on preprocessing, specifically canonicalization, followed by sentence classification (for XML and JSON) or file type recognition (for PDF). The bottleneck is partially attributed to the word tokenization and POS tagging in the Stanford CoreNLP API. One way to mitigate this problem is to use the Stanford CoreNLP Server[6]. Alternatively, we can use Stanza (Qi et al., 2020), the successor of Stanford CoreNLP. Empirical results have shown that it is faster than CoreNLP in several NLP tasks. Stanza was written in Python, but we can develop a RESTful service. The slowness can also be attributed to the poor garbage collection in Java, which can impact CPU usage massively. A more systematic and fine-grained profiling is needed to diagnose the root cause of this problem.

8 Conclusions and Future Works

We developed SmartCiteCon (SCC), a Java API to extract explicit and implicit citation context from academic literature. The API implements an SVM model achieving an $F_1 = 85.6\%$. SCC accepts XML (in PLoS ONE schema or GROBID schema), PDF, and JSON (in S2ORC schema) formats. The output of SCC is a JSON file containing marked citation contexts and paper metadata if available. We applied SCC on the PMC subset of the CORD-19 dataset and obtained about 11.8 million citation context sentences in which 10.2 million are implicit citation context.

One limitation of SCC is that the model was trained on papers in computational linguistics, so more careful evaluation and feature distribution analysis should be performed when applying the model to other domains. In the future, we will explore word embedding models to enrich semantic features and improve scalability by overcoming performance bottlenecks.

[5] `https://jats.nlm.nih.gov/`

[6] `https://stanfordnlp.github.io/CoreNLP/corenlp-server.html`

Acknowledgments

We thank Zikun Feng for setting up a web-based user interface and Shengwei Lei for constructive discussion.

References

Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1435–1446.

Arman Cohan and Nazli Goharian. 2015. Scientific article summarization using citation-context and article's discourse structure. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 390–400, Lisbon, Portugal. Association for Computational Linguistics.

Isaac Councill, C Lee Giles, and Min-Yen Kan. 2008. ParsCit: an open-source CRF reference string parsing package. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Shengwei Lei, Haihua Chen, Yong Huang, and Wei Lu. 2016. Research on automatic recognition of academic citation context. *Library and Information Service*, 60(17).

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Patrice Lopez. 2009. GROBID: combining automatic bibliographic data recognition and term extraction for scholarship publications. In *Proceedings of the 13th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL'09, pages 473–474, Berlin, Heidelberg. Springer-Verlag.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Jonathan Murray. 2015. Finding implicit citations in scientific publications. Master's thesis, KTH Royal Insitute of Technology.

Hidetsugu Nanba and Manabu Okumura. 1999. Towards multi-paper summarization using reference information. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, IJCAI '99, page 926–931, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Vahed Qazvinian and Dragomir R. Radev. 2010. Identifying non-explicit citing sentences for citation-based summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 555–564, Uppsala, Sweden. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 101–108. Association for Computational Linguistics.

Ulrich Schäfer and Benjamin Weitz. 2012. Combining OCR outputs for logical document structure markup. technical background to the ACL 2012 contributed task. In *Proceedings of the Special Workshop on Rediscovering 50 Years of Discoveries@ACL 2012, Jeju Island, Korea, July 10, 2012*, pages 104–109. Association for Computational Linguistics.

K. Sugiyama, T. Kumar, M. Kan, and R. C. Tripathi. 2010. Identifying citing sentences in research papers using supervised learning. In *2010 International Conference on Information Retrieval Knowledge Management (CAMP)*, pages 67–72.

Synthetic vs. Real Reference Strings for Citation Parsing, and the Importance of Re-training and Out-Of-Sample Data for Meaningful Evaluations: Experiments with GROBID, GIANT and CORA

Mark Grennan
Trinity College Dublin, School of Computer Science and Statistics, Ireland
grennama@tcd.ie

Joeran Beel
University of Siegen, Department of Electrical Engineering and Computer Science, Germany
joeran.beel@uni-siegen.de

Abstract

Citation parsing, particularly with deep neural networks, suffers from a lack of training data as available datasets typically contain only a few thousand training instances. Manually labelling citation strings is very time-consuming, hence, synthetically created training data could be a solution. However, as of now, it is unknown if synthetically created reference-strings are suitable to train machine learning algorithms for citation parsing. To find out, we train Grobid, which uses Conditional Random Fields, with a) human-labelled reference strings from 'real' bibliographies and b) synthetically created reference strings from the GIANT dataset. We find[1] that both synthetic and organic reference strings are equally suited for training Grobid (F1 = 0.74). We additionally find that retraining Grobid has a notable impact on its performance, for both synthetic and real data (+30% in F1). Having as many types of labelled fields as possible during training also improves effectiveness, even if these fields are not available in the evaluation data (+13.5% F1). We conclude that synthetic data is suitable for training (deep) citation parsing models. We further suggest that in future evaluations of reference parsing tools, both evaluation data being similar and data being dissimilar to the training data should be used to obtain more meaningful results.

1 Introduction

Accurate citation data is needed by publishers, academic search engines, citation & research-paper recommender systems and others to calculate impact metrics (Nisa Bakkalbasi et al., 2006; Jacso, 2008), rank search results (Beel and Gipp, 2009a,b), generate recommendations (Beel et al., 2016; Eto, 2019; Färber et al., 2018; Färber and Jatowt, 2020; Jia and Saule, 2018; Livne et al., 2014) and other applications e.g. in the field of bibliometric-enhanced information retrieval (Cabanac et al., 2020). Citation data is often parsed from unstructured bibliographies found in PDF files on the Web (Figure 1). To facilitate the parsing process, a dozen (Tkaczyk et al., 2018a) open source tools were developed including ParsCit (Councill et al., 2008), Grobid (Lopez, 2009, 2013), and Cermine (Tkaczyk et al., 2015). Grobid is typically considered the most effective one (Tkaczyk et al., 2018a). There is ongoing research that continuously leads to novel citation-parsing algorithms including deep learning algorithms (An et al., 2017; Bhardwaj et al., 2017; Nasar et al., 2018; Prasad et al., 2018; Rizvi et al., 2019; Rodrigues Alves et al., 2018; Zhang, 2018) and meta-learned ensembles (Tkaczyk et al., 2018c,b).

Most parsing tools apply supervised machine learning (Tkaczyk et al., 2018a) and require labelled training data. However, training data is rare compared to other disciplines where datasets may have millions of instances. To the best of our knowledge, existing citation-parsing datasets typically contain a few thousand instances and are domain specific (Figure 2). This may be sufficient for traditional machine learning algorithms but not for deep learning, which shows a lot of potential for citation parsing (An et al., 2017; Bhardwaj et al., 2017; Nasar et al., 2018; Prasad et al., 2018; Rizvi et al., 2019). Even for traditional machine learning, existing datasets may not be ideal as they often lack diversity in terms of citation styles.

Recently, we published GIANT, a synthetic dataset with nearly 1 billion annotated reference strings (Grennan et al., 2019). More precisely, the dataset contains 677,000 unique reference strings, each in around 1,500 citation styles (e.g. APA,

[1]The work presented in this manuscript is based on Mark Grennan's Master thesis "1 Billion Citation Dataset and Deep Learning Citation Extraction" at Trinity College Dublin, Ireland, 2018/2019

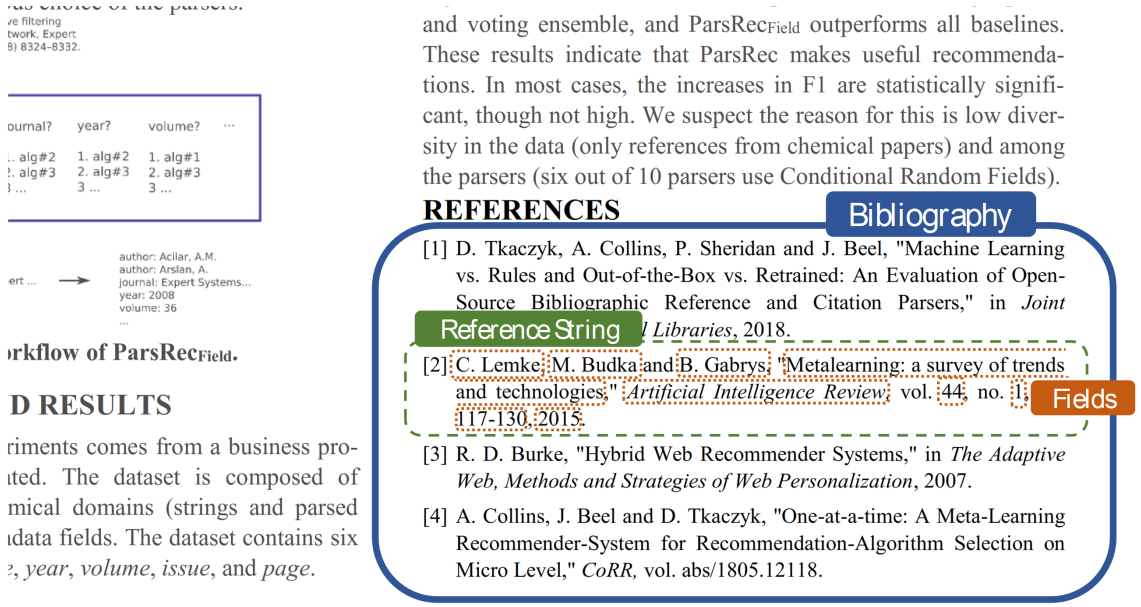


Figure 1: Illustration of a 'Bibliography' with four 'Reference Strings', each with a number of 'Fields'. A reference parser receives a reference string as input, and outputs labelled fields, e.g. authors="C. Lemke ..."; title="Metalearning: a survey ..."; ...

Harvard, ACM). The dataset was synthetically created. This means, the reference strings are not 'real' reference strings extracted from 'real' bibliographies. Instead, we downloaded 677,000 references in XML format from CrossRef, and used Citeproc-JS (Frank G. Bennett, 2011) with 1,500 citation styles to convert the 677,000 references into a total of 1 billion annotated citation strings (1,500 * 677,000).[2]

We wonder how suitable a synthetic dataset like GIANT is to train machine learning models for citation parsing. Therefore, we pursue the following research question:

1. How will citation parsing perform when trained on synthetic reference strings, compared to being trained on real reference strings?

Potentially, synthetic data could lead to higher citation parsing performance, as synthetic datasets may contain more data and more diverse data (more citation styles). Synthetic datasets like GIANT could potentially also advance (deep) citation parsing, which currently suffers from a lack of 'real' annotated bibliographies at large scale.

In addition to the above research question, we aimed to answer the following questions:

2. To what extent does citation-parsing (based on machine learning) depend on the amount of training data?

3. How important is re-training a citation parser for the specific data it should be used on? Or, in other words, how does performance vary if the test data differs (not) from the training data?

4. Is it important to have many different fields (author, year, ...) for training, even if the fields are not available in the final data?

2 Related Work

We are aware of eleven datasets (Figure 2) with annotated reference strings. The most popular ones are probably Cora and CiteSeer. Researchers also often use variations of PubMed. Several datasets are from the same authors, and many datasets include data from other datasets. For instance, the Grobid dataset is based on some data from Cora, PubMed, and others (Lopez, 2020). New data is continuously added to Grobid's dataset. As such, there is not "the one" Grobid dataset. GIANT (Grennan et al., 2019) is the largest and most diverse dataset in terms of citation styles, but GIANT is, as mentioned, synthetically created.

Cora is one of the most widely used datasets but has potential shortcomings (Anzaroot and Mc-

[2] We use the terms 'citation parsing', 'reference parsing', and 'reference-string parsing' interchangeably.

Callum, 2013; Councill et al., 2008; Prasad et al., 2018). Cora is homogeneous with citation strings only from Computer Science. It is relatively small and only has labels for "coarse-grained fields" (Anzaroot and McCallum, 2013). For example, the author field does not label each author separately. Prasad et al. conclude that a "shortcoming of [citation parsing research] is that the evaluations have been largely limited to the Cora dataset, which is [...] unrepresentative of the multilingual, multidisciplinary scholastic reality" (Prasad et al., 2018).

3 Methodology

To compare the effectiveness of synthetic vs. real bibliographies, we used Grobid. Grobid is the most effective citation parsing tool (Tkaczyk et al., 2018a) and the most easy to use tool based on our experience. Grobid uses conditional random fields (CRF) as machine learning algorithm. Of course, in the long-run, it would be good to conduct our experiments with different machine learning algorithms, particularly deep learning algorithms, but for now we concentrate on one tool and algorithm. Given that all major citation-parsing tools – including Grobid, Cermine and ParsCit – use CRF we consider this sufficient for an initial experiment. Also, we attempted to re-train Neural ParsCit (Prasad et al., 2018) but failed doing so, which indicates that the ease-of-use of the rather new deep-learning methods is not yet as advanced as the established citation parsing tools like Grobid.

We trained Grobid, the CRF respectively, on two datasets. $\text{Train}_{\text{Grobid}}$ denotes a model trained on 70% (5,460 instances) of the dataset that Grobid uses to train its out-of-the box version. We slightly modified the dataset, i.e. we removed labels for 'pubPlace', 'note' and 'institution' as this information is not contained in GIANT, and hence a model trained on GIANT could not identify these labels[3]. $\text{Train}_{\text{GIANT}}$ denotes the model trained on a random sample (5,460 instances) of GIANT's 991,411,100 labeled reference strings. Our expectation was that both models would perform similar, or, ideally, $\text{Train}_{\text{GIANT}}$ would even outperform $\text{Train}_{\text{Grobid}}$.

To analyze how the amount of training data affects performance, we additionally trained $\text{Train}_{\text{GIANT}}$, on 1k, 3k, 5k, 10k, 20k, and 40k instances of GIANT.

We evaluated all models on four datasets. $\text{Eval}_{\text{Grobid}}$ comprises of the remaining 30% of Grobid's dataset (2,340 reference strings). $\text{Eval}_{\text{Cora}}$ denotes the Cora dataset, which comprises, after some cleaning, of 1,148 labelled reference strings from the computer science domain. $\text{Eval}_{\text{GIANT}}$ comprises of 5,000 random reference strings from GIANT.

These three evaluation datasets are potentially not ideal as evaluations are likely biased towards one of the two trained models. Evaluating the models on $\text{Eval}_{\text{GIANT}}$ likely favors $\text{Train}_{\text{GIANT}}$ since the data for both $\text{Train}_{\text{GIANT}}$ and $\text{Eval}_{\text{GIANT}}$ is highly similar, i.e. it originates from the same dataset. Similarly, evaluating the models on $\text{Eval}_{\text{Grobid}}$ likely favors $\text{Train}_{\text{Grobid}}$ as $\text{Train}_{\text{Grobid}}$ was trained on 70% of the original Grobid dataset and this 70% of the data is highly similar to the remaining 30% that we used for the evaluation. Also, the Cora dataset is somewhat biased, because Grobid's dataset contains parts of Cora. We therefore created another evaluation dataset.

$\text{Eval}_{\text{WebPDF}}$ is our 'unbiased' dataset with 300 manually annotated citation strings from PDFs that we collected from the Web. To create $\text{Eval}_{\text{WebPDF}}$, we chose twenty different words from the homepages of some universities[4]. Then, we used each of the twenty words as a search term in Google Scholar. From each of these searches, we downloaded the first four available PDFs. Of each PDF, we randomly chose four citation strings. This gave approximately sixteen citation strings for each of the twenty keywords. In total, we obtained 300 citation strings. We consider this dataset to be a realistic, though relatively small, dataset for citation parsing in the context of a web-based academic search engine or recommender system.

We measure performance of all models with precision, recall, F1 (Micro Average) and F1 (Macro Average) on both field level and token level. We only report 'F1 Macro Average on field level' as all metrics led to similar results.

All source code, data (including the WebPDF dataset), images, and an Excel sheet with all results (including precision and recall and token level results) is available on GitHub

[3]This is a shortcoming of GIANT. However, the purpose of our current work is to generally compare 'real' vs synthetic data. Hence, both datasets should be as similar as possible in terms of available fields to make a fair comparison. Therefore, we removed all fields that were not present in both datasets.

[4]The words were: bone, recommender systems, running, war, crop, monetary, migration, imprisonment, hubble, obstetrics, photonics, carbon, cellulose, evolutionary, revolutionary, paleobiology, penal, leadership, soil, musicology.

Dataset Name	# Instances	Domain
Cora [29]	1,295	Computer Science
CiteSeer [16]	1,563	Artificial Intelligence
Umass [2]	1,829	STEM
FLUX-CiM CS [20]	300	Computer Science
FLUX-CiM HS [20]	2,000	Health Science
GROBID [26–28]	6,835	Multi-Domain (Cora, arXiv, PubMed...)
PubMed (Central) [9, 17]	Varies	Biomedical
GROTOAP2 (Cermine) [35–37]	6,858	Biomedical & Computer Science
CS-SW [20]	578	Semantic Web Conferences
Venice [33]	40,000	Humanities
GIANT [19]	991 million	Multi-Domain (~1,500 Citation Styles)

Figure 2: List of Citation Datasets

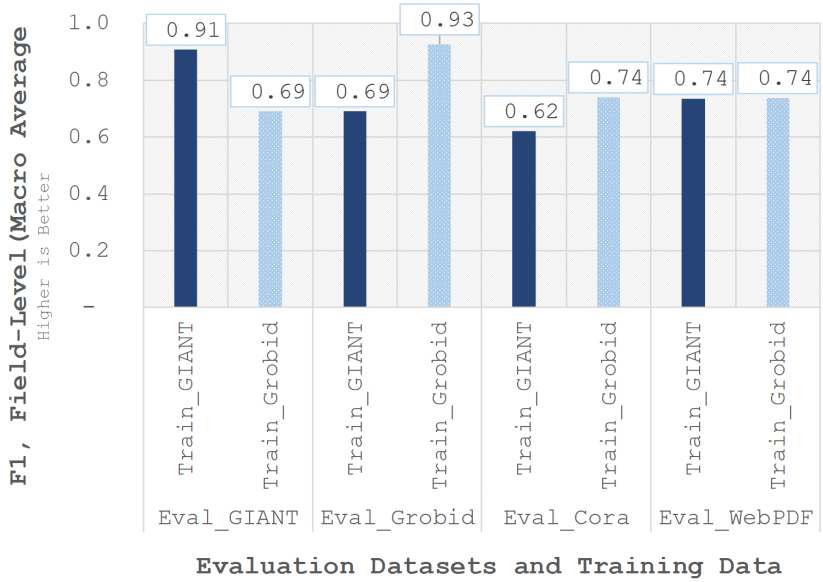


Figure 3: F1 of the two models ($\text{Train}_{\text{Grobid}}$ and $\text{Train}_{\text{GIANT}}$) on the four evaluation datasets.

https://github.com/BeelGroup/GIANT-The-1-Billion-Annotated-Synthetic-Bibliographic-Reference-String-Dataset/.

4 Results

The models trained on Grobid ($Train_{Grobid}$) and GIANT ($Train_{GIANT}$) perform as expected when evaluated on the three 'biased' datasets $Eval_{Grobid}$, $Eval_{Cora}$ and $Eval_{GIANT}$ (Figure 3). When evaluated on $Eval_{Grobid}$, $Train_{Grobid}$ outperforms $Train_{GIANT}$ by 35% with an F1 of 0.93 vs. 0.69. When evaluated on $Eval_{GIANT}$, results are almost exactly the opposite: This time, $Train_{GIANT}$ outperforms $Train_{Grobid}$ by 32% with an F1 of 0.91 vs. 0.69. On $Eval_{Cora}$, the difference is less strong but still notable. $Train_{Grobid}$ outperforms $Train_{GIANT}$ by 19% with an F1 of 0.74 vs. 0.62. This is not surprising as Grobid's training data includes some Cora data.

While these results generally might not be surprising, they imply that both synthetic and real data lead to very similar results and 'behave' similarly when used to train models that are evaluated on data being (not) similar to the training data.

Also interesting is the evaluation on the WebPDF dataset. The model trained on synthetic data ($Train_{GIANT}$) and the model trained on real data ($Train_{Grobid}$) perform alike with an F1 of 0.74 each (Figure 3)[5]. In other words, synthetic and human-labelled data perform equally well for training our machine learning models.

Looking at the data in more detail reveals that some fields are easier to parse than others (Figure 4). For instance, the 'date' field (i.e. year of publication) has a constantly high F1 across all models and evaluation datasets (min=0.86; max=1.0). The 'author' field also has a high F1 throughout all experiments (min=0.75; max=0.99). In contrast, parsing 'booktitle' and 'publisher' seems to strongly benefit from training based on samples similar to the evaluation data. When evaluation and training data is highly similar (e.g. $Train_{GIANT}$–$Eval_{GIANT}$ or $Train_{Grobid}$–$Eval_{Grobid}$), F1 is relatively high (typically above 0.7). If the evaluation data is different (e.g. $Train_{GIANT}$– $Eval_{Grobid}$), F1 is low (0.15 and 0.16 for $Train_{Grobid}$ and $Train_{GIANT}$ respectively on $Eval_{WebPDF}$). The difference in F1 for parsing the book-title is around factor 6.5, with an F1 of 0.97 ($Train_{Grobid}$) and 0.15 respectively ($Train_{GIANT}$) when evaluated on $Eval_{Grobid}$.

Similarly, F1 for parsing the book-title on $Eval_{GIANT}$ differs by around factor 3 with an F1 of 0.75 ($Train_{GIANT}$) and 0.27 ($Train_{Grobid}$) respectively. While it is well known, and quite intuitive, that different fields are differently difficult to parse, we are first to show that field accuracy varies for different fields differently depending on whether or not the model was trained on data (not) being similar to the evaluation data.

In a side experiment, we trained a new model $Train_{Grobid}$+ with additional labels for institution, note and pubPlace (those we removed for the other experiments). $Train_{Grobid}$+ outperformed $Train_{Grobid}$ notably with an F1 of 0.84 vs. 0.74 (+13.5%) when evaluated on $Eval_{WebPDF}$. This indicates that the more fields are available for training, the better the parsing of all fields becomes even if the additional fields are not in the evaluation data. This finding seems plausible to us and confirms statements by Anzaroot and McCallum but, to the best of our knowledge, we are first to quantify the benefit. It is worth noting that citation parsers do not always use the same fields (Figure 6). For instance, Cermine extracts relatively few fields, but is one of few tools extracting the DOI field.

Our assumption that more training data would generally lead to better parsing performance – and hence GIANT could be useful for training standard machine learning algorithms – was not confirmed. Increasing training data from 1,000 to 10,000 instances improved F1 by 6% on average over the four evaluation datasets (Figure 5). More precisely, increasing data from 1,000 to 3,000 instances improved F1, on average, by 2.4%; Increasing from 3,000 to 5,000 instances improved F1 by another 2%; Increasing further to 10,000 instances improved F1 by another 1.6%. However, increasing to 20,000 or 40,000 instances leads to no notable improvement, and in some cases even to a decline in F1 (Figure 5).

5 Summary and Discussion

In summary, both models – one trained on synthetic data (GIANT) and one trained on 'real' human-annotated reference strings (Grobid) – performed very similar. On the main evaluation dataset (WebPDF) both models achieved an F1 of 0.74. Similarly, if a model was evaluated on data different from its training data, F1 was between 0.6 and

[5] All results are based on the Macro Average F1. Looking at the Micro Average F1 shows a slightly better performance for $Train_{Grobid}$ than for $Train_{GIANT}$ (0.82 vs. 0.80), but the difference is neither large nor statistically significant ($p¡0.05$).

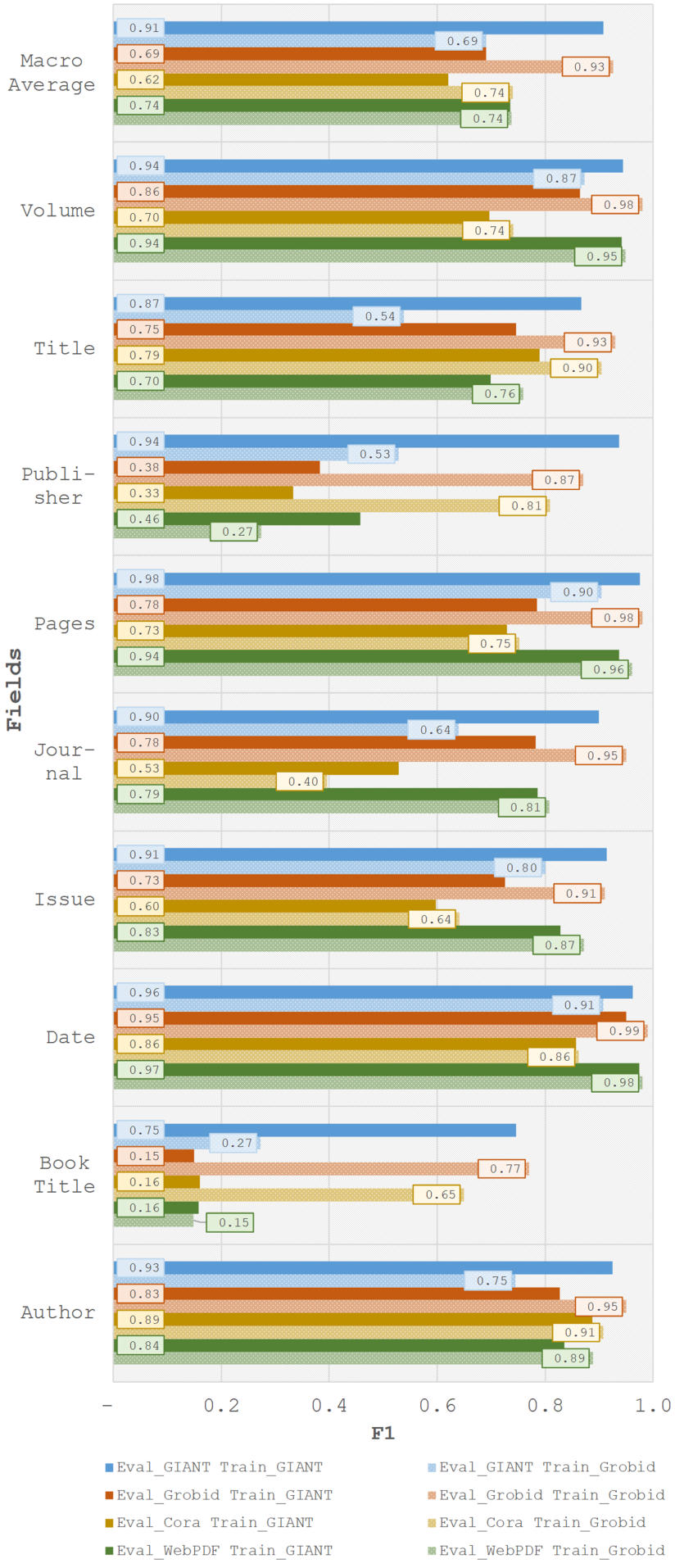


Figure 4: F1 for different fields (title, author, ...), evaluation dataset and training data.

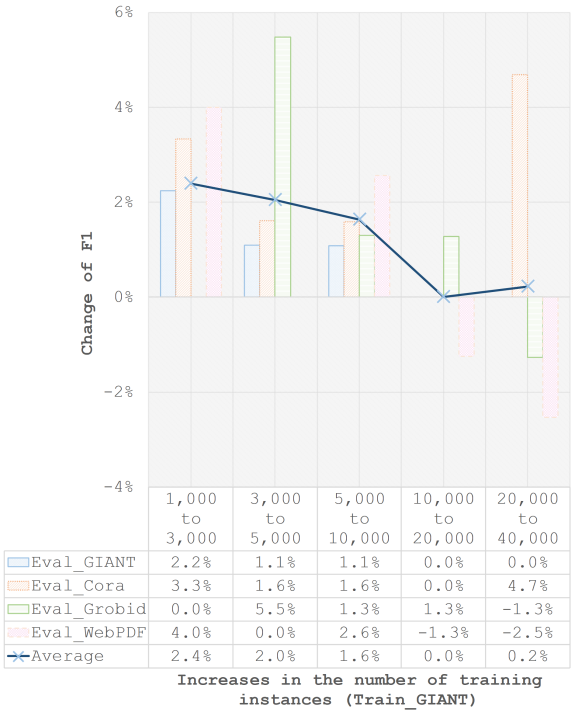


Figure 5: Performance (F1) of $\text{Train}_{\text{GIANT}}$ on the four evaluation datasets, by the number of training instances.

Citation Parser	Approach	Extracted Fields
Biblio	Regular Expressions	author, date, editor, genre, issue, pages, publisher, title, volume, year
BibPro	Template Matching	author, title, venue, volume, issue, page, date, journal, booktitle, techReport
CERMINE	Machine Learning (CRF)	author, issue, pages, title, volume, year, DOI, ISSN
GROBID		authors, booktitle, date, editor, issue, journal, location, note, pages, publisher, title, volume, web, institution
ParsCit		author, booktitle, date, editor, institution, journal, location, note, pages, publisher, tech, title, volume
Neural ParsCit	Deep Learning	author, booktitle, date, editor, institution, journal, location, note, pages, publisher, tech, title, volume

Figure 6: The approach and extracted fields of six popular open-source citation parsing tools

0.7. If a model was evaluated on data similar to the training data, F1 was above 0.9 (+30%). F1 only increased up to a training size of around 10,000 instances (+6% compared to 1,000 instances). Additional fields (e.g. pubplace) in the training data increased F1 notably (+13.5%), even if these additional fields were not in the evaluation data.

These results lead us to the following conclusions.

First, there seems to be little benefit in using synthetic data (e.g. GIANT (Grennan et al., 2019)) for training traditional machine learning models (i.e. conditional random fields). The existing datasets with a few thousand training instances seem sufficient.

Second, citation parsers should, if possible, be (re)trained on data that is similar to the data that should actually be parsed. Such a re-training increased performance by around 30% in our experiments. This finding may also explain why researchers often report excellent performance of their tools and approaches with e.g. F1's of over 0.9. These researchers typically evaluate their models on data highly similar to the training data. This might be considered a realistic scenario for those cases when re-training is possible. However, reporting such results creates unrealistic expectations for scenarios without the option to re-train, i.e. for users who just want to use a citation parser like Grobid out-of-the-box. Therefore, we propose that future evaluations of citation parsing algorithms should be conducted on at least two datasets: One dataset that is similar to the training dataset, and one out-of-sample dataset that differs from the training data.

Third, citation parsers should be trained with as many labelled field types as possible, even if these fields will not be in the data that should be parsed. Such a fine-grained training improved F1 by 13.5% in our experiments.

Fourth, having ten times as much training data (10,000 vs. 1,000) improved the parsing performance by 6%, without notable improvements beyond 10,000 instances. Annotating a few thousand instances should be feasible for many scenarios. Hence, businesses and organizations who want the maximum accuracy should annotate their own data for training as this likely will lead to large increases in accuracy (+30%, see conclusion 3).

Fifth, given how similar synthetic and traditionally annotated data perform, synthetic data likely is suitable to train deep neural networks for citation parsing. This, of course, has yet to be empirically to be shown. However, if our assumption holds true, deep citation parsers could greatly benefit

from synthetic data like GIANT.

For the future, we see the need to extend our experiments to different machine learning algorithms and datasets (e.g. unarXive (Saier and Färber, 2020) or CORE (Knoth and Zdrahal, 2012)). It would also be interesting to analyze if and to what extend synthetic data could improve related disciplines. This may include citation-string matching, i.e. analyzing whether two different reference strings refer to the same document (Ghavimi et al., 2019), or the extraction of mathematical formulae (Greiner-Petter et al., 2020) or titles (Lipinski et al., 2013) from scientific articles.

Acknowledgments

We are grateful for the support received by Martin Schibel, Andrew Collins and Dominika Tkaczyk in creating the GIANT dataset. We would also like to acknowledge that this research was partly conducted with the financial support of the ADAPT SFI Research Centre at Trinity College Dublin. The ADAPT SFI Centre for Digital Media Technology is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106.

References

Dong An, Liangcai Gao, Zhuoren Jiang, Runtao Liu, and Zhi Tang. 2017. Citation metadata extraction via deep neural network-based segment sequence labeling. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1967–1970. ACM.

Sam Anzaroot and Andrew McCallum. 2013. A new dataset for fine-grained citation field extraction. *ICML Workshop on Peer Reviewing and Publishing Models*.

Nisa Bakkalbasi, Kathleen Bauer, Janis Glover, and Lei Wang. 2006. Three options for citation tracking: Google Scholar, Scopus and Web of Science. *Biomedical Digital Libraries*, 3.

Joeran Beel and Bela Gipp. 2009a. Google Scholar's Ranking Algorithm: An Introductory Overview. In *Proceedings of the 12th International Conference on Scientometrics and Informetrics (ISSI'09)*, volume 1, pages 230–241, Rio de Janeiro (Brazil). International Society for Scientometrics and Informetrics. Available at http://docear.org.

Joeran Beel and Bela Gipp. 2009b. Google Scholar's Ranking Algorithm: The Impact of Citation Counts (An Empirical Study). In *Proceedings of the 3rd IEEE International Conference on Research Challenges in Information Science (RCIS'09)*, pages 439–446, Fez (Morocco). IEEE. Available at http://docear.org.

Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. 2016. Research paper recommender systems: A literature survey. *International Journal on Digital Libraries*, (4):305–338.

Akansha Bhardwaj, Dominik Mercier, Andreas Dengel, and Sheraz Ahmed. 2017. Deepbibx: Deep learning for image based bibliographic data extraction. In *International Conference on Neural Information Processing*, pages 286–293. Springer.

Guillaume Cabanac, Ingo Frommholz, and Philipp Mayr. 2020. Bibliometric-enhanced information retrieval (bir) 10th anniversary workshop edition. *arXiv preprint arXiv:2001.10336*.

I.G. Councill, C.L. Giles, and M.Y. Kan. 2008. ParsCit: An open-source CRF reference string parsing package. In *Proceedings of LREC*, volume 2008, pages 661–667. European Language Resources Association (ELRA).

Masaki Eto. 2019. Extended co-citation search: Graph-based document retrieval on a co-citation network containing citation context information. *Information Processing & Management*, 56(6):102046.

Michael Färber and Adam Jatowt. 2020. Citation recommendation: Approaches and datasets. *arXiv preprint arXiv:2002.06961*.

Michael Färber, Alexander Thiemann, and Adam Jatowt. 2018. Citewerts: A system combining citeworthiness with citation recommendation. In *European Conference on Information Retrieval*, pages 815–819. Springer.

Jr. Frank G. Bennett. 2011. The citeproc-js Citation Processor.

Behnam Ghavimi, Wolfgang Otto, and Philipp Mayr. 2019. An evaluation of the effect of reference strings and segmentation on citation matching. In *International Conference on Theory and Practice of Digital Libraries*, pages 365–369. Springer.

André Greiner-Petter, Moritz Schubotz, Fabian Müller, Corinna Breitinger, Howard S Cohl, Akiko Aizawa, and Bela Gipp. 2020. Discovering mathematical objects of interest–a study of mathematical notations. *arXiv preprint arXiv:2002.02712*.

Mark Grennan, Martin Schibel, Andrew Collins, and Joeran Beel. 2019. Giant: The 1-billion annotated synthetic bibliographic-reference-string dataset for deep citation parsing. In *27th AIAI Irish Conference on Artificial Intelligence and Cognitive Science*, pages 101–112.

P. Jacso. 2008. Testing the calculation of a realistic h-index in Google Scholar, Scopus, and Web of Science for FW Lancaster. *Library Trends*, 56(4):784–815.

Haofeng Jia and Erik Saule. 2018. Graph embedding for citation recommendation. *arXiv preprint arXiv:1812.03835*.

Petr Knoth and Zdenek Zdrahal. 2012. Core: three access levels to underpin open access. *D-Lib Magazine*, 18(11/12).

Mario Lipinski, Kevin Yao, Corinna Breitinger, Joeran Beel, and Bela Gipp. 2013. Evaluation of header metadata extraction approaches and tools for scientific pdf documents. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries (JCDL'13)*, pages 385–386.

Avishay Livne, Vivek Gokuladas, Jaime Teevan, Susan T Dumais, and Eytan Adar. 2014. Citesight: supporting contextual citation recommendation using differential search. pages 807–816.

Patrice Lopez. 2009. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *International conference on theory and practice of digital libraries*, pages 473–474. Springer.

Patrice Lopez. 2013. Grobid, github repository. *https://github.com/kermitt2/grobid/*.

Patrice Lopez. 2020. Training data query #535. *GitHub https://github.com/kermitt2/grobid/issues/535*.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2018. Information extraction from scientific articles: a survey. *Scientometrics*, 117(3):1931–1990.

Animesh Prasad, Manpreet Kaur, and Min-Yen Kan. 2018. Neural parscit: a deep learning-based reference string parser. *International Journal on Digital Libraries*, 19(4):323–337.

Syed Tahseen Raza Rizvi, Andreas Dengel, and Sheraz Ahmed. 2019. Deepbird: An automatic bibliographic reference detection approach. *arXiv preprint arXiv:1912.07266*.

Danny Rodrigues Alves, Giovanni Colavizza, and Frédéric Kaplan. 2018. Deep reference mining from scholarly literature in the arts and humanities. *Frontiers in Research Metrics and Analytics*, 3:21.

Tarek Saier and Michael Färber. 2020. unarxive: a large scholarly data set with publications' full-text, annotated in-text citations, and links to metadata. *Scientometrics*, pages 1–24.

Dominika Tkaczyk, Andrew Collins, Paraic Sheridan, and Joeran Beel. 2018a. Machine learning vs. rules and out-of-the-box vs. retrained: An evaluation of open-source bibliographic reference and citation parsers. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, JCDL '18, pages 99–108, New York, NY, USA. ACM.

Dominika Tkaczyk, Rohit Gupta, Riccardo Cinti, and Joeran Beel. 2018b. Parsrec: A novel meta-learning approach to recommending bibliographic reference parsers. In *Proceedings of the 26th Irish Conference on Artificial Intelligence and Cognitive Science (AICS)*, volume 2259, pages 162–173. CEUR-WS.

Dominika Tkaczyk, Paraic Sheridan, and Joeran Beel. 2018c. Parsrec: A meta-learning recommender system for bibliographic reference parsing tools. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys)*.

Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Łukasz Bolikowski. 2015. Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(4):317–335.

Yiqing Zhang. 2018. Towards highly accurate publication information extraction from academic homepages.

Term-Recency for TF-IDF, BM25 and USE Term Weighting

Divyanshu Marwah
School of Computer Science
and Statistics
Trinity College Dublin
Dublin, Ireland
marwahd@tcd.ie

Joeran Beel
School of Computer Science
and Statistics
Trinity College Dublin
Dublin, Ireland
joeran.beel@scss.tcd.ie

Abstract

Effectiveness of a recommendation in an Information Retrieval (IR) system is determined by relevancy scores of retrieved results. Term weighting is responsible for computing the relevance scores and consequently differentiating between the terms in a document. However, current term weighting formula like TF-IDF weigh terms only based on term frequency and inverse document frequency irrespective of other important factors. This results in uncertainty in cases when both TF and IDF values are same for more than one document, hence resulting in same term weight values. In this paper, we propose a modification of TF-IDF and other term-weighting schemes that weights terms additionally based on the recency of a term, i.e. the metric based on the year the term occurred for the first time and the document frequency. We modified the term weighting schemes TF-IDF, BM25 and Universal Sentence Encoder (USE) to additionally consider the recency of a term and evaluated them on three datasets. Our modified TF-IDF outperformed the standard TF-IDF on all three datasets; the modified USE outperformed the standard USE on two of the three datasets; the modified BM25 did not outperform the standard BM25 term-weighting scheme.

1 Introduction

Term Weighting is one of the most crucial tasks in information retrieval and recommender systems. It is method of quantifying terms in a document to determine the importance of the words in the document and the corpus (El-Khair, 2009). Apart from recommendation engine and information retrieval, term weighting is effective in many scenarios such as text mining, text classification, duplicate image detection (Chum et al., 2008), document clustering, and even in medical science research. In text categorization and data mining, efficient. term weighting brings a considerable boost in effectiveness (Domeniconi et al., 2015). Several term weighting approaches are used in different applications basically derived from the frequency and distribution of words in documents (Domeniconi et al., 2015).

TF-IDF is one of the classic term weighting approaches, that is most frequently used and was found to be used, for instance, by 83% of text-based research paper recommender systems (Beel et al., 2017). TF-IDF as the name suggests, is made up of two parts, term frequency (TF) and inverse document frequency (IDF). TF gives the number of times a term occurs in a document. The basis is that the more frequently a term occurs, the more it is important for the context of the document (Beel et al., 2017). IDF is computed as the inverse frequency of documents containing the searched term. The idea behind this is that a rare term should be given higher importance as compared to frequently occurring terms such as articles, pronouns, etc. There have been numerous researches on TF-IDF, and many extensions and alternatives are suggested. Some other term weighting models used are BM25, LM Dirichlet, Divergence from independence, etc. Text and sentence embedding models such as Universal Sentence Encoder (USE) (Cer et al., 2018), Google's BERT, InferSent, etc are also used in text classification tasks. These different approaches depend on the type, size of corpus, types of queries, and they use different term metrics to determine the effectiveness of term in a document and corpus.

In case of information retrieval task, there are certain limitations in standard term weighting approaches. Analyzing the simple approach of TF-IDF, that weights term based on the frequency distribution in the corpus. The real issue in this method is the assumption that frequency distribution remains constant with time, without contemplating the diverse contexts for different terms. In short periods, this holds, however, over longer time this assumption fails. For example, consider two

terms, ”COVID19” and ”neural networks”, that have different origin years. Now, there are probably fewer documents containing the term ”COVID19” than documents containing the term ”neural networks”, simply because ”COVID19” is a relatively new term, while ”neural networks” is a term being used since decades. However, they would be weighted similarly without considering the difference in the origins. The issue that terms have temporal distributions of frequency, not just space distribution is unaccounted when using the standard term weighting methodologies.

Considering this uncertainty in term weighting, we suggest a time-normalized term weighting approach, which reflects the age of a term. As the vocabulary changes over time, our intuition is to identify a term's age based on its first usage and current year and distinguish between the documents based on the age of the terms used. Hence, we propose to weigh terms not only on their frequency distributions but also temporal distributions. Furthermore, we demonstrate the significance of adding a time-based feature by comparing our method with state-of-the-art baseline models, that is, TF-IDF, BM25 and USE embedding. Experimental results show substantial improvements over the baseline models for similar recommendations.

2 Related Work

TF-IDF is a relatively old approach and there have been many studies comparing the results of TF-IDF with other states of the art term weighting schemes. Also, different researches have suggested novel variants and enhancing algorithms solving various issues. For instance, (Beel et al., 2017) points out the lack of personalization in classic TF-IDF. The authors have highlighted the issue of access to the document corpus for calculating IDF and another issue of ignoring the information from the user's document collection for recommendations and user modelling. Thus, a novel term weighting is suggested, that does not require the document corpus and uses the user's document collection for user modelling.

In another paper, (Domeniconi et al., 2015) points out the problem of using IDF in text classification. The basic idea behind IDF is that a term occurring frequently has negligible distinguishing power, however, in the case of text classification, this might not be true, because, highly frequent terms in different documents of the same category can be helpful in text classification. Hence, the authors suggested a supervised learning approach to calculate IDF excluding the category under consideration.

(Park et al., 2005) suggests a novel approach to term weighting based on the term positions along with the TF and IDF terms. The authors studied the term patterns that occur in the documents using the wavelet transform method. The paper also suggests that the documents are ranked more relevant if the query terms are close to each other.

Utilizing temporal feature has also proved to be an efficient way for recommenders and time normalized recommendations are certainly receiving growing application in recent times (Campos et al., 2014). One of the researches (Kacem et al., 2014), suggests usage of time-normalized term weighting for user modelling. The authors have used the time of social/web search of terms to form the short and long-term contexts and further creating a user profile based on the same. The comparative study of this algorithm with the standard TF-IDF suggests a significant improvement in results centered on the time normalized user models. Considering this research of temporal context's effect on term weights, we propose a Time Normalized TF-IDF algorithm for information retrieval and recommender system, discussed and implemented in this paper.

3 Time Normalized Term Weighting

In a classic term weighting approach, the terms are weighted irrespective of the different contexts or usage or recency. In this paper, we try to emphasize on the importance of term recency in relevant results retrieval. The premise for this algorithm is that, if a term is devised newly then there are probably a lesser number of documents containing the term compared to the term which is being used for a longer duration of time. In this algorithm, we introduce a time factor along with the regular TF-IDF values. This time-based factor is formulated from the origin year of the word and the document frequency of the term giving the metric as documents per year. For a given term w in document $d \epsilon D$, where D is the document corpus D with size N, term-age is calculated as:

$$t_{w,D} = \log(df_{w,D}/(y_{diff} + 1))^{1}, \qquad (1)$$

[1]This is an updated formula with an added 1 in the denominator, for our experiments, we used the older version of formula

where y_{diff} is calculated as :

$$y_{diff} = y_{current} - y_{origin} \quad (2)$$

where y_{origin} is the year of first usage of the word and $Y_{current}$ is the present year. This current year remains constant in the calculations, giving us the sort of age for the word. We take the logarithm of the terms to normalize the value, since this can go to a large number based on the size of the corpus. Also, we take up the absolute value of log, so that we don't have negative weight values. The y_{origin} can be traced from multiple places depending on the problem statement. For example, if a research paper recommender is being developed, the origin year can be retrieved as the year of first occurrence of the term in the recommendation corpus. Or in case of web search, time of first search of the term can be used. Likewise, for some instances the terms can be traced to their etymology and the year of first occurrence can be fetched. Now the updated formula for term weight calculation for tTF-IDF is given as:

$$wt_{w,d} = t_{w,D} * tf_{w,d} * log(\frac{N}{df_{w,d}}) \quad (3)$$

where $t_(w, D)$ is the time-factor calculated value in equation (1) $tf_(w, d)$ is the number of times term w occurs in a document d, and $df_(w, D)$ is the number of documents in which w appears in D.

Likewise, in case of time normalized BM25 (tBM25) model, the term age, $t_(w, D)$ is multiplied to the classic BM25 formula for the time normalized model. For USE embedding approach, cosine similarity is used to calculate the term weights. In the time normalized model, we multiply the term age factor, $t_(w, D)$ with the cosine similarity function to get the updated time normalized USE (tUSE) model.

Now, assume that a term is new and occurs in reasonable number of documents, then the value of $t_(w, D)$ will be large and hence the term weight will be large. Similarly, if the term is being used for many years and is occurring in many documents, it will relatively reduce the value of the time-factor, thus giving it low importance.

A caution which needs to be taken while implementing this algorithm is to check for more commonly occurring non relevant terms which are normalized by using IDF should not get boosted. This can be taken care of while calculating the value of y_{origin}, and such terms can be ignored so they don't boost up the term weights based on non-relevant terms.

4 Implementation

4.1 Data

4.1.1 TREC Washington Post Corpus (Post, 2018)

This collection contains 608,180 news articles and blog posts, along with 50 queries from TREC – 2018 news background linking task (Soboroff et al., 2018), and expected set of results. For the purpose of testing our hypothesis, we use a sample of 20909 documents with approximately 2400 relevant documents. However, this has been done only for time-based index due to scalability and resource constraints. And the term age is still calculated considering the entire corpus and does not affect the algorithmic logic. The relevant fields in this dataset are id, URL, title, author, and article text.

4.1.2 Web Answer Passage(WebAP) Dataset (Keikha et al., 2014, 2015)

This collection contains 8027 articles from the web, which are answers to 82 TREC queries. The dataset contains the following fields: unique document id, target question id, and passage. The results contain 50 relevant documents, given as question_id, document number and relevance as ranked from 1 to 50.

4.1.3 CiteULike Dataset (Wang et al., 2013)

This dataset is collected from CiteULike and Google Scholar and contains 17013 documents with the following fields: document id, title of the research paper, and abstract. We are given another file in this dataset, that contains the referenced articles for every document. We have randomly selected 116 test topics having exactly 10 citations to be used as our ground truth.

4.2 Architecture/Methodology

We implemented text-based recommendation systems using the data mentioned in the last section. This is implemented using TF-IDF, BM25 and USE embedding models. Further, we devised an algorithm to calculate $t_(w, D)$ as described earlier. And scoring is done using customized plugins. Finally, we compare the results of different algorithms using the evaluation metrics described later.

The first index is created with the same mapping structure as given in the input dataset files. For

the second index, we add a time normalized term weight parameter as a payload to the terms while indexing the documents. A third index is created using the time normalized index for USE embedding model. In this index, we calculate the term vectors using the pre-trained TensorFlow model (Cer et al., 2018) and store it in a 512-length vector field. This methodology remains the same for all the datasets. Following steps are used for calculations of term age:

- Consider the article text of the document and fetch the origin year for every word from etymonline.com
- Calculate the difference in number of years from the year of first occurrence, to the current year. We have assumed the base year for our corpus to be 2017(TREC News), 2015(Web AP) and 2019(CiteULike) since that is the year of latest publications. This has been done for uniform term weighting across the corpus.
- Now the term weight is calculated using the formula given in section 3.

An important part to note is, that every term is not given a term weight, this happens if the origin year of the term is not traceable, or the terms are most frequently used such as articles, or prepositions. We have used the following evaluation metrics to evaluate the significance of retrieved results:

- *Precision @10*: We have calculated the precision value for top 10 fetched results on the given set of input queries and take an average of the results for comparison.
- *Recall*: For calculating the recall, we have considered the queries having less than 100 results, in case of TREC news. And then recall is calculated as the number of relevant retrieved document divided by the number of relevant documents present in the index. For other datasets, since the number of relevant results is fixed, so the value of precision and recall remains the same.
- *F1 Score*: Since F1 score uses both the precision and recall values, so for calculating the precision scores, we have used the same results from the recall measure and used a fixed denominator as the number of retrieved results. Formula used for F1 score is given as:

$$F1 = 2 * \frac{Precision * Recal}{Precision + Recall} \quad (4)$$

- *Normalized Discounted Cumulative Gain*: DCG is calculated at specific rank position p, given by

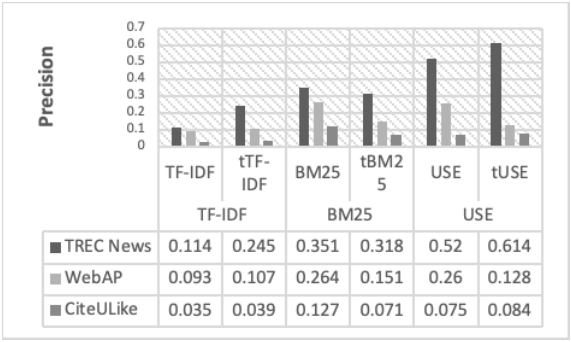
$$DCG_p = \sum_{i=1}^{p} rel_i / \log_2(i+1) \quad (5)$$

where, rel_i is the relevance score of a document at position i. And NDCG is calculated by considering the DCG of ideal order along with the DCG values and is given by

$$nDCG_p = DCG_p / IDCG_p \quad (6)$$

5 Results and Discussion

In 2 out of 3 algorithms, our term-recency modification improved the performance notably. When measured by p@10, tTF-IDF outperformed TF-IDF by an average 47% and tUSE outperformed USE in 2 of the 3 datasets by 14.3% but performed 50% worse in the other dataset (Figure 1). The time normalized BM25 version, however, performed 32% worse than BM25. NDCG@10 leads to similar results (Figure 2). For CiteULike dataset, NDCG cannot be calculated, since there is no ranking specified for the citations.



	TF-IDF	tTF-IDF	BM25	tBM25	USE	tUSE
TREC News	0.114	0.245	0.351	0.318	0.52	0.614
WebAP	0.093	0.107	0.264	0.151	0.26	0.128
CiteULike	0.035	0.039	0.127	0.071	0.075	0.084

Figure 1: P@10 comparison

On closer analysis of the BM25 model, we see 27 out of 50 queries gave better or almost similar results in case of time normalized model when compared to the classic approach. These results are also promising and need to be worked upon for better results in the future work.

For calculating the recall and F1 scores, we fetch the top 100 results for the given query sets. For

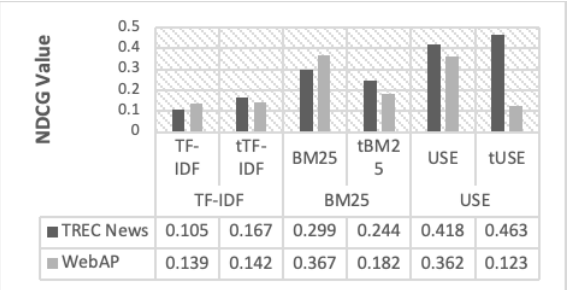


Figure 2: NDCG@10 comparison

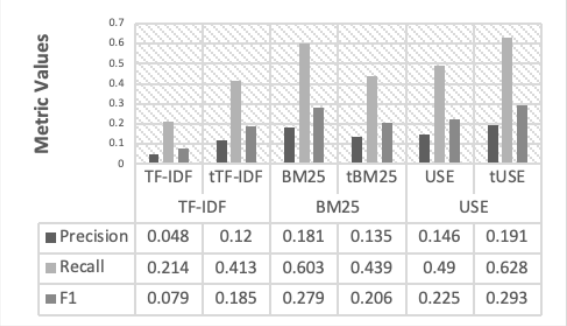


Figure 3: Precision, Recall, F1@100 scores for TREC News

uniformity in the metric calculation, we compute the precision scores as well. The result metrics are shown in Figure 3. We see a 150% improvement in the tTF-IDF model over TF-IDF and a 31% improvement in tUSE model over the USE model. Recall and F1 scores for WebAP and CiteULike would be same as precision scores since the number of relevant results in dataset remains the same, so they are not shown.

Analyzing the tUSE model in WebAP dataset, we see it does not perform well against the USE model. One of the possible reasons for this might be the size of the corpus used, that is, TREC news corpus has approximately 600k documents while Web AP dataset has just 6k documents, which is 100 times less than the former dataset. However, this is an inference based on the results retrieved and has not been verified. There might be other possible reasons, such as the size of documents, size of queries used, number of proper nouns in the queries, etc. Or probably term age might not be a relevant metric for this dataset. These possible reasons still need to be analyzed before affirming out a conclusion on these contrasting results.

6 Conclusion

In this paper, we proposed a novel algorithm for term weighting. The presented approach shows the significance of temporal distribution along with existing space distribution of terms. We suggest the scheming of a term recency parameter based on the origin of the word and the usage in the document corpus. This factor is used along with the standard weighting values (such as TF and IDF) for relevance scoring in the information retrieval system. The algorithm is tested on a news dataset, with queries trying to find links with the documents, Web answer retrieval dataset and research papers citations dataset. We have also extended the algorithm to other text embedding models that are BM25 and USE.

Experiments conducted on the IR system show that term-recency based TF-IDF and tUSE model outperforms the classic TF-IDF and classic USE algorithms with a significant margin when measured in terms of average precision, recall, F1 and NDCG. It has set up a strong premise for our ongoing research on ways to improve recommendation effectiveness. Future works for this can be to find ways to improve the time-based BM25 model and testing the algorithm's performance in other tasks such as text classification, and user modelling. Furthermore, we also plan to test different normalization factors for calculating the term age and then using it in the scoring algorithm.

Acknowledgments

We would like to thank Sneha Srivastava, Rachit Rastogi and Shivam Khanduri for their comments, suggestions and feedback.

References

Joeran Beel, Stefan Langer, and Bela Gipp. 2017. *TF-IDuF: A novel term-weighting scheme for user modeling based on users' personal document collections*. University of Illinois.

Pedro G. Campos, Fernando Díez, and Iván Cantador. 2014. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1-2):67–119.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, and Chris Tar. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Ondrej Chum, James Philbin, and Andrew Zisserman. 2008. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, volume 810, pages 812–815.

Giacomo Domeniconi, Gianluca Moro, Roberto Pasolini, and Claudio Sartori. 2015. A study on term weighting for text categorization: A novel supervised variant of tf. idf. In *DATA*, pages 26–37.

Ibrahim Abu El-Khair. 2009. *Term Weighting*, pages 3037–3040. Springer US, Boston, MA.

Ameni Kacem, Mohand Boughanem, and Rim Faiz. 2014. Time-sensitive user profile for optimizing search personlization. In *International conference on user modeling, adaptation, and personalization*, pages 111–121. Springer.

Mostafa Keikha, Jae Hyun Park, W. Bruce Croft, and Mark Sanderson. 2014. Retrieving passages and finding answers. In *Proceedings of the 2014 Australasian Document Computing Symposium*, pages 81–84.

Mostafa Keikha, Jae Hyun Park, W. Bruce Croft, and Mark Sanderson. 2015. Web answer passages (webap) dataset.

Laurence A. F. Park, Kotagiri Ramamohanarao, and Marimuthu Palaniswami. 2005. A novel document retrieval method using the discrete wavelet transform. *ACM Transactions on Information Systems (TOIS)*, 23(3):267–298.

Washington Post. 2018. Trec washington post corpus.

Ian Soboroff, Shudong Huang, and Donna Harman. 2018. Trec 2018 news track overview. In *The Twenty-Seventh Text RE-trieval Conference (TREC 2018) Proceedings*.

Hao Wang, Binyi Chen, and Wu-Jun Li. 2013. Collaborative topic regression with social regularization for tag recommendation. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

The Normalized Impact Index for Keywords in Scholarly Papers to Detect Subtle Research Topics

Daisuke Ikeda
Kyushu University
Moto-oka 744, Fukuoka,
819-0395, Japan

Yuta Taniguchi
Kyushu University
Moto-oka 744, Fukuoka,
819-0395, Japan

Kazunori Koga
Kyushu University
Moto-oka 744, Fukuoka,
819-0395, Japan
National Institutes of Natural Sciences
Toranomon 4-3-13, Tokyo,
105-0001, Japan

Abstract

Mainly due to the open access movement, the number of scholarly papers we can freely access is drastically increasing. A huge amount of papers is a promising resource for text mining and machine learning. Given a set of papers, for example, we can grasp past or current trends in a research community. Compared to the trend detection, it is more difficult to forecast trends in the near future, since the number of occurrences of some features, which are major cues for automatic detection, such as the word frequency, is quite small before such a trend will emerge. As a first step toward trend forecasting, this paper is devoted to finding subtle trends. To do this, the authors propose an index for keywords, called *normalized impact index*, and visualize keywords and their indices as a heat map. The authors have conducted case studies using some keywords already known as popular, and we found some keywords whose frequencies are not so large but whose indices are large.

1 Introduction

Thanks to the recent open access movement, we can freely access to a huge amount of papers on scholarly repositories, such as institutional repositories maintained by academic institutions. According to IRUS-UK,[1] there exits about 2M items on more than 200 repositories in the UK, as of May 2020. According to NII,[2] there exist more than 2.4M full-text papers on 734 institutional repositories in Japan, as of March 2020. In addition to institutional repositories, we also have disciplinary repositories, such as arXiv.[3]

We can also use a global aggregation servie, which collects papers on repositories. For example, CORE [4] collects papers from more than one thousand data providers in about 150 countries, and provides search APIs, dump files, and search facility for collected papers (Knoth and Zdrahal, 2012). The latest dump file provided by CORE contains 123M metadata items, 85.6M abstracts, and 9.8M full text papers. Some commercial publishers also began to provide APIs for automatic processing.[5]

Basically, items on scholarly repositories are readable PDF files. When research results were published on paper, research papers were final outcomes of the researches. In case of digital media, however, contents of the papers can be an input for automatic processing. We can find many researches which use scholarly papers as input for computer algorithms. For example, some entities, like dataset names, used in papers are automatically extracted (Ikeda and Seguchi, 2017; Ikeda and Taniguchi, 2019), and papers are used to predict research impacts of a new given paper (Baba et al., 2019) and to predict new materials (Tshitoyan et al., 2019).

The final goal of our research is to forecast popular trends in the near future. A typical method for this is to use a clustering algorithm, which is unsupervised learning, and divides target items into groups based on a predefined distance metric. Some approaches use clustering algorithms to divide words in papers into groups, such as the topic model (Griffiths and Steyvers, 2004; Bolelli et al., 2009). Once we introduce a distance metric to data, a target data item is defined as a point in the space defined by the metric, and thus we can compare similarities between any two points. In this sense, this approach uses an absolute distance. There also exit relative approaches, like network structures, in which we know that two items are adjancent. In par-

[1] https://irus.jisc.ac.uk/
[2] https://www.nii.ac.jp/irp/en/archive/statistic/
[3] https://arxiv.org/
[4] https://core.ac.uk/
[5] https://www.elsevier.com/about/policies/text-and-data-mining

ticular, we can naturally construct multiple network structures from papers, like networks of authors, citations, words, and their combinations (Duvvuru et al., 2012; Salatino et al., 2017). However, these researches assume that there are already a number of publications (Salatino et al., 2018). In this sense, these approaches are for topic detection, not for topic forecast.

In this paper, we try to find small topics as a first step toward forecasting future topics. To this end, we propose an index for keywords to measure their impact, assuming a keyword denotes a research topic. We use a relative frequency in the definition of the index to find small topics. As far as the authors know, the frequency of keywords is not directly used to detect topics in research papers, unlike topic or trend detection in general text data. The authors think that this is because a frequency based method requires a list of stop words to remove unnecessary keywords, but it is too costly to construct it for each discipline in case of research papers.

To evaluate the proposed index, we use some popular keywords in one discipline, and we check if the proposed indices for them can grasp their popularity. Using this approach, we do not have to consider the issue of stop words. In other words, we try to find some properties among popular topics with the proposed index. For comparison, we also show topic detection by absolute frequency and a standard clustering algorithm.

2 Normalized Impact Index

We assume the range of publication years, $y_1, y_2, \ldots, y_N$, and let $Y = \{y_1, y_2, \ldots, y_N\}$. For $y \in Y$, $D(y)$ denotes the set of papers published in y.

For a word w and a year $y \in Y$, the *normalized impact index*, denoted by $h(w, y)$, is defined as as follows:

$$h(w, y) = \frac{f(w, y)}{|D(y)| \sum_{t=y_1}^{y_N} f(w, t)},$$

where $f(w, y)$ is the number of occurrences (frequencies) of w in $D(y)$.

The proposed index for w and y is a relative frequency, normalized by both the number of publications in y and the total frequency of w among all years. Therefore, we can compare $h(w_1, y_1)$ and $h(w_2, y_2)$.

To understand the meaning of the index, let us assume that $|D(y)| = 1$ tentatively. Then we can treat $h(w, y)$ as a probability since we have $\sum_y h(w, y) = 1$. So, when we depict this index as a bar chart for some w whose height is $h(w, y_i)$, the total area of the bars for w is normalized to 1. Therefore, we can compare any two words w_1 and w_2, in the view point of their trends.

When we consider trends of keywords, it is natural to see temporal changes of the index from some reference year y_1, that is,

$$h(w, y) - h(w, y_1), \tag{1}$$

where $y > y_1$ for $y \in Y - \{y_1\}$. For some $y(\neq y_1)$, if $h(w, y) - h(w, y_1) > 0$ (resp. < 0), then the relative usage of w in y becomes larger (resp. smaller) than that in y_1. This leads to a heat map of the proposed index for keywords.

3 Case Study

In this section, we apply the proposed index to a real dataset to confirm its efficacy. As described in Section 1, a frequency based method suffers from the issue of stop words. To avoid the issue, we check the values of the proposed index for some keywords the authors selected from some specific field. These keywords are already known as popular topics. Therefore, it means that we only check positive examples.

Since the proposed index is defined with relative frequencies, we show the result of topic detection with absolute frequencies for comparison (see Section 3.2). Then, we apply a clustering algorithm to our dataset in Section 3.3, to confirm that a clustering algorithm for keywords can find large topics, not small ones as described in Section 1.

3.1 Dataset

We use a set of abstracts, not the whole papers, from 2000 to 2018, obtained by searching "plasma chemical vapor deposition" at Web of Science. The number of abstracts we obtained is 69,384.

In addition to stop words of English, we also removed tokens starting or ending with special symbols, such as "[" and "+". Then we converted capital letters to lower-case ones.

3.2 Topic Detection by Frequency

As the first case study, we check if a method based on frequency can find a potentially popular topic.

Figure 1 contains four graphs, showing the numbers of papers found by queries at Web of Science. One common line is contained in all graphs

in Figure 1, which is the number of papers found by "plasma chemical vapor deposition". In other

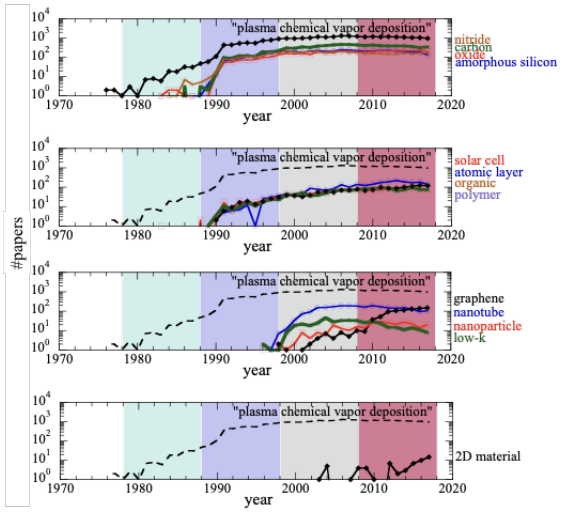


Figure 1: Each graph shows the change of the number of papers found by the corresponding query with "plasma chemical vapor deposition", such as "nitride plasma chemical vapor deposition", as the publication year advances (some data originally from Fig. 6 and 7 in (Iwase et al., 2019)).

words, this line shows the year-by-year changes of the number of papers containing this query. We call the line for this query the *base line* of this field.

Each of the other lines shows the number of papers found by "plasma chemical vapor deposition" plus the corresponding keyword. For example, the red line in the top graph is obtained by "oxide plasma chemical vapor deposition". These searches are search within the original query, and thus these lines are below the base line. One of the authors chose these additional keywords, based on the heat map in Figure 2 in addition to his expertise. Basically, they are known to be popular topics.

In the four graphs, an upper graph contains keywords whose frequencies are larger. In the top graph of "nitride", "carbon", "oxide", and "amorphous silicon", we see that these keywords are large topics in this field and the shapes of graphs are similar to the base line. Compared to the top graph, the second one contains smaller topics, but they have emerged in early 90s, and increased its publications steadily.

Compared to the two top graphs, keywords for the other two graphs are relatively new topics, and thus the numbers of papers containing these topics are much smaller. In particular, the number of the papers about "2D material", meaning 2 dimensional materials, is quite small. In spite of its small frequency, this topic has potential to be big in this field because "2D material" is a more conceptual word than "graphene", which is a 2D material, and the Nobel Prize was awarded to researchers studied graphene in 2010.

Therefore, methods based on the frequency of a keyword can not find such a trend at very early stages.

3.3 Topic Detection by Clustering

Next, we consider a clustering algorithm as a method to find research topics.

For a clustering algorithm, we used Non-negative Matrix Factorization (NMF), which decomposes a given matrix $\boldsymbol{V}$ into two matrices $\boldsymbol{W}\boldsymbol{H}$, where all emelements in those matrices are required to be non-negative (Lee and Seung, 1999).

Using the set of abstracts, we can construct a term-document matrix $\boldsymbol{V}$, where w_{ij} is the frequency for the ith term in the jth document, that is the jth document d_j has $w_{1j}, w_{2j}, \ldots$ as its elements.

Let D and V be the number of documents and one of vocabularies, respectively. Then, the size of $\boldsymbol{V}$ is $D \times V$. When we apply NMF to $\boldsymbol{V}$, we have to specify a parameter K, which defines the sizes of two matrices: $D \times K$ and $K \times V$ for $\boldsymbol{W}$ and $\boldsymbol{H}$.

We can see $\boldsymbol{W}$ as a weight matrix and $\boldsymbol{H}$ as a base matrix, and an original document is expressed as a weighted linear combination of base elements. In this expression, we can see that a base matrix consits of K base vectors.

Table 1 shows the top 10 keywords with largest weights for each base vector, where we set $K = 10$. There exist K topics, each of which has 10 keywords with the top 10 largest weights in the topic.

From this table, we can find many major topics in this field. For example, the first cluster contains "chemical vapor deposition", and the second and 10th ones "carbon nanotubes" and "thin film", respectively, both of which are major materials used in this field. However, we can not find minor topics from this decomposition.

3.4 Topic Detection by the Proposed Index and Heat Map

In this section, we detect topics using the normalized impact index and its visualization.

No.	The top 10 keywords with largest weights in a topic
1	deposition, chemical, vapor, rate, process, high, gas, using, PECVD, pressure
2	carbon, growth, nanotubes, CNTs, field, emission, electron, catalyst, grown, chemical
3	silicon, layer, solar, amorphous, cells, layers, chemical, cell, nitride, high
4	films, deposited, thin, properties, spectroscopy, optical, amorphous, content, using, x-ray
5	surface, surfaces, roughness, layer, chemical, contact, treatment, energy, morphology, atomic
6	plasma, power, gas, density, treatment, enhanced, using, pressure, hydrogen, discharge
7	C, degrees, temperature, annealing, growth, substrate, temperatures, si, low, rights
8	diamond, growth, microwave, substrate, CVD, high, nucleation, quality, substrates, grown
9	coatings, coating, properties, DLC, chemical, using, deposited, wear, elsevier, reserved
10	film, thin, thickness, substrate, deposited, stress, structure, dielectric, nm, ratio

Table 1: The top 10 keywords with largest weights in a topic found by NMF.

Figure 2 shows a heat map, defined by (1), for keywords in our dataset. One column corresponds

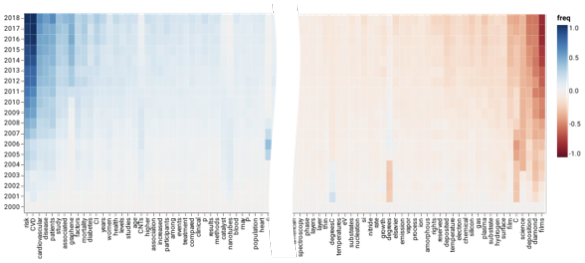


Figure 2: The heat map shows values of (1) for each keyword extracted from our dataset, where one column corresponds to one keyword, and a cell in the column indicates the value of (1).

to one keyword, and each row to one year. We only show the left and right parts of the heat map because the original figure is too wide since there are many keywords.

Each cell shows the difference between the normalized impact index of that year and the reference year, 2000, for some word. That is, it shows the value of (1), where blue (resp. red) cells are positive (resp. negative) values, meaning the relative frequency of the corresponding year for the word is larger (resp. smaller) than that of the reference year.

Figure 3 shows temporal changes of the proposed indices for some selected keywords, some of which appear in Figure 1 and the other ones are chosen from the heat map.

"graphene", "2D", "nanotube", "low-k" (low dielectric constant), "h-BN" (hexagonal boron nitride), and "GaAs" are names of materials, and "interconnect" and "fuel" are the keywords of the plasma chemical vapor deposition (CVD for short) applications, where "interconnect" refers as inter-

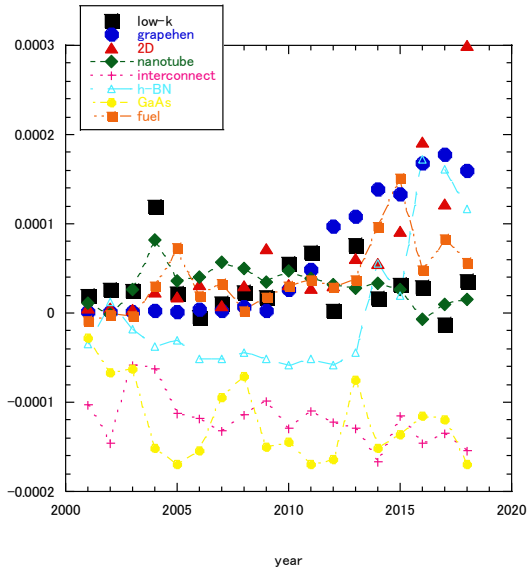


Figure 3: The graph shows the temporal change of the proposed index for some keywords, such as "low-k".

connect in semiconductor devices and "fuel" as fuel cells.

For interconnect, the proposed index was negative and decreased from 2000. Plasma CVD as interconnect process technology has been losing interest. The proposed index for fuel increases continuously and there was temporary booming in 2000 and 2015.

Both "nonotube" and "low-k" appeared in the third graph of Figure 1. From this graph, we can see sharp rises of their frequencies. However, from the proposed index for these keywords, we can not say these topics are actively examined in papers.

As shown in Figure 1, "2D" has its small frequency although it has potential to be a big trend because unique characteristics of 2D materials have been found then the research of 2D materials seems

to become active as the trigger of the graphene Nobel Prize. On the other hand, the proposed index for "2D" rises sharply in Figure 3.

The index for "h-BN" has negative values until 2012, which seems to have lost the interest of researchers, but after that it increases rapidly. In fact, "h-BN" has been studied as a 2D semiconductor material recently. In this sense, "h-BN" can be seen as a 2D material family, and so it is convincing the sharp rise for "h-BN".

4 Conclusion

In this paper, we have introduced an index to find keywords, which express small topics, using relative frequencies. As visualization, the difference of the proposed index from the reference year, 2000 in this paper, is depicted as a heat map. Therefore, we can easily find subtle topics even if their absolute frequencies are not so large. We have conducted case studies using the proposed index, and confirmed that some keywords, which are already known as popular, show sharp rises of the proposed index.

As described in Section 3, we have only checked popular keywords. So it is an important future work to check all keywords whose values of the proposed index.

Even if we find some keywords with high values of the proposed index, you might want to check their absolute frequencies. Therefore, it is also important to develop a visualization tool which enables to check both the absolute frequency and the proposed index. Similarly, it is an important future work for the tool to introduce a grouping facility, which groups a different keywords in a hierarchical way, and then we can grasp transitions of topics with flexible granularity with the tool. To do so, we can use some vocabulary system, like one in (Salatino et al., 2019), or word embeddings to measure the distances between two keywords.

Acknowledgments

In this paper, the authors used data from Web of Science, a product of Clarivate.

References

Takahiro Baba, Kensuke Baba, and Daisuke Ikeda. 2019. Citation Count Prediction using Abstracts. *Journal of Web Engineering*, 18(1–3):207–228.

Levent Bolelli, Şeyda Ertekin, and C. Lee Giles. 2009. Topic and Trend Detection in Text Collections Using Latent Dirichlet Allocation. In *Advances in Information Retrieval (ECIR 2009)*, Lecture Notes in Artificial Intelligence 5478, pages 776–780.

Arjun Duvvuru, Sagar Kamarthi, and Sivarit Sultornsanee. 2012. Undercovering research trends: Network analysis of keywords in scholarly articles. In *Proceedings of Ninth International Conference on Computer Science and Software Engineering*, pages 265–270.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding Scientific Topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235.

Daisuke Ikeda and Daisuke Seguchi. 2017. Automatically Extracting Keywords from Documents for Rich Indexes of Searchable Data Repositories. In *Proceedings of the 12th International Conference of Open Repositories*.

Daisuke Ikeda and Yuta Taniguchi. 2019. Toward Automatic Identification of Dataset Names in Scholarly Articles. In *Developments in Open Science and Research Data Management: 8th International Conference on Data Science and Institutional Research*.

Taku Iwase, Yoshito Kamaji, Song Yun Kang, Kazunori Koga, Nobuyuki Kuboi, Moritaka Nakamura, Nobuyuki Negishi, Tomohiro Nozaki, Shota Nunomura, Daisuke Ogawa, Mitsuhiro Omura, Tetsuji Shimizu, Kazunori Shinoda, Yasushi Sonoda, Haruka Suzuki, Kazuo Takahashi, Takayoshi Tsutsumi, Kenichi Yoshikawa, Tatsuo Ishijima, and Kenji Ishikawa. 2019. Progress and perspectives in dry processes for emerging multidisciplinary applications: how can we improve our use of dry processes? *Japanese Journal of Applied Physics*, 58(SE).

Petr Knoth and Zdenek Zdrahal. 2012. CORE: Three Access Levels to Underpin Open Access. *D-Lib Magazine*, 18(11/12).

Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791.

Angelo Salatino, Francesco Osborne, Thiviyan Thanapalasingam, and Enrico Motta. 2019. The CSO Classifier: Ontology-Driven Detection of Research Topics in Scholarly Articles. In *Proceedings of the 23rd International Conference on Theory and Practice of Digital Libraries*, Lecture Notes in Computer Science 11799.

Angelo A. Salatino, Francesco Osborne, and Enrico Motta. 2017. How are topics born? understanding the research dynamics preceding the emergence of new areas. *PeerJ Computer Science*, 3(e119).

Angelo A. Salatino, Francesco Osborne, and Enrico Motta. 2018. AUGUR: Forecasting the Emergence of New Research Topics. In *Proceedings of the*

18th ACM/IEEE on Joint Conference on Digital Libraries, pages 303–312.

Vahe Tshitoyan, John Dagdelen, Leigh Weston, Alexander Dunn, Ziqin Rong, Olga Kononova, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. 2019. Unsupervised Word Embeddings Capture Latent Knowledge from Materials Science Literature. *Nature*.

Representing and Reconstructing PhySH: Which Embedding Competent?

Xiaoli Chen
National Science Library
University of Chinese Academy of Sciences
Chinese Academy of Sciences
chenxl@mail.las.ac.cn

Zhixiong Zhang
National Science Library
Wuhan Library
University of Chinese Academy of Sciences
Chinese Academy of Sciences
Zhangzhx@mail.las.ac.cn

Abstract

In this paper, we conduct a comprehensive comparison of well-known embeddings' capability in capturing the hierarchical Physics knowledge. Several key findings are: (i) Poincaré embeddings do outperform if trained on PhySH, but it fails if trained on co-occurrence pairs which are extracted from raw text. (ii) No algorithm can properly learn hierarchies from the more realistic case of co-occurrence pairs, which contains more noisy relations other than hierarchical relations. (iii) Our statistic analysis of Poincaré embedding's representation of PhySH shows successful hierarchical representation share two characteristics: firstly, upper-level terms have a smaller semantic distance to root; secondly, upper-level hypernym-hyponym pairs should be further apart than lower-level hypernym-hyponym pairs.

1 Introduction

Concept hierarchy or taxonomy[1] is highly organized and expertly curated hierarchical hypernym-hyponym sets. How to effectively represent these terms with the hierarchical relation is the main hurdle for automatically taxonomy construction and other downstream applications.

Though embeddings have been taken for granted in most NLP pipelines, none of the previous work has fully explored which embeddings can capture hierarchical scientific knowledge. Even though Poincaré embedding is proved to have a better ability to capture hierarchical relations, it is learned based on existing WordNet hypernym-hyponym pairs. It is never been tested in the scientific domain. In this paper, we conduct a comprehensive comparison of well-known embeddings' performance in reconstructing Physical Subject Headings (PhySH) from raw APS datasets.

[1]In this paper, we use *taxonomy* and *concept hierarchy* as equal term.

Our main contributions are mainly three-fold: Firstly, for the first time, we compare mainstream embeddings' capability to represent and reconstruct Physical Subject Headings (PhySH) both from raw text and PhySH. Secondly, our experiment shows Poincaré embedding is not sufficient for taxonomy induction from raw text. Thirdly, we explore the characteristics of successful representation of PhySH, which might be the inspiration for better taxonomy construction algorithms.

2 Related Work

Representations for Concept Hierarchy. Representations for concept hierarchy has been receiving quite growing interests in recent years (Kozareva et al., 2008; Carlson et al., 2010; Shen et al., 2018). It is the basis of automaticly taxonomy construction. In the survey study of (Wang et al., 2017), there are *Pattern-based* (Hearst, 1992; Wu et al., 2012; Kozareva and Hovy, 2010) methods and *distributional* (Navigli and Velardi, 2004; Luu et al., 2014; Padó and Lapata, 2003; Baroni and Lenci, 2010; Nguyen et al., 2017) methods use hand-crafted rule-based, co-occurrence features,syntactic features or graph features to learn representations of hierarchical pairs. They also apply pretrained neural laguage models such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014).

Recently, Poincaré embedding (Nickel and Kiela, 2017) is proposed to better represent hierarchical relations. Following works like (Law et al., 2019) use Lorentzian distance to replace the Poincaré metric, (Dhingra et al., 2018) extends Poincaré embedding to apply in raw text with re-parameterization technique, (Leimeister and Wilson, 2019) and (Tifrea et al., 2019) introduce hyperbolic embeddings in word embeddings like Skipgram and GloVe. Effectively in reconstructing WordNet though, the Poincaré embedding is not quite perfect yet (De Sa et al., 2018). It has only been tested on WordNet re-

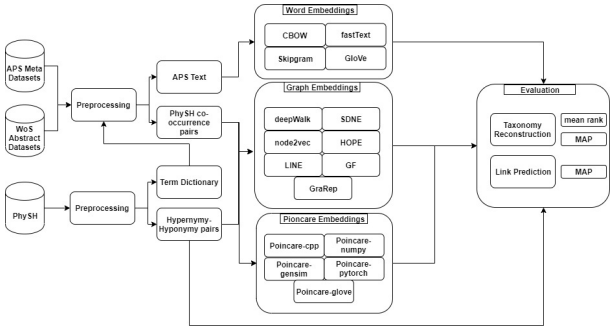


Figure 1: Evaluation Pipeline

construction and Hyperlex entailment (Nickel and Kiela, 2017). Whether it is an effective tool in representing hierarchical relations from raw domain text need to be further explored.

Embeddings Analysis. With the fast pacing of text representation technology, it is also important to revisit existing embedding methods for different downstream tasks. Several previous works have explicitly done this work based on their unique perspectives.Gladkova et al. (2016) explores GloVe's ability to encode different morphological and semantic relations. (Zuccon et al., 2015) analyze word embeddings for information retrieval. Nooralahzadeh et al. (2019) compared COW and Skipgram by using Gensim implementation (Řehůřek and Sojka, 2010) with several different hyper-parameters settings and different domain corpus. Sanchez and Riedel (2017) explored different datasets in evaluation hypernyms identification by using GloVe. (Lastra-DÃ■az et al., 2019) surveys main word embeddings for word similarity.

Despite the above-mentioned work, there is still a missing part describing which embedding is the optimal choice for taxonomy induction. In this paper, we design our evaluation pipeline to choose the optimal embedding scheme for taxonomy learning and construction. In our paper, we consider two perspectives to represent and construct concept hierarchy: (i) Learn and construct from raw texts by word embeddings; (ii) Learn and construct from extracted co-occurrence pairs from raw texts by graph embeddings and Poincaré embeddings.

3 Method

In our pipeline (Figure 1), we follow three steps: raw text and PhySH preprocessing; learn various embeddings with different hyperparameters; evaluate embeddings by reconstructing PhySH.

We evaluate the following embeddings:

	Model Name	Metric	Dimensions					
			5	10	20	50	100	200
Word Embeddings	GloVe	*mean rank*	2168·30	2568·93	2237·33	2142·94	2188·83	2271·32
		MAP	0·18	0·05	0·06	0·07	0·06	0·06
	COW	*mean rank*	2883·64	3196·44	2937·09	3162·85	1894·02	3096·82
		MAP	0·69	0·63	0·70	0·63	**0·72**	0·64
	Skipgram	*mean rank*	2595·87	3939·61	3091·23	2732·08	3683·61	2893·45
		MAP	0·68	0·60	0·67	0·68	0·63	0·70
	fastText	*mean rank*	2461·97	3004·28	2903·78	3391·16	3456·85	2493·87
		MAP	0·67	0·67	0·69	0·66	0·59	0·65
Graph Embeddings	deepWalk	*mean rank*	**244·03**	469·47	624·21	726·95	780·47	811·30
		MAP	0·18	0·05	0·06	0·05	0·05	0·05
	GF	*mean rank*	1189·78	1003·12	916·56	825·82	682·73	629·40
		MAP	0·01	0·01	0·01	0·01	0·02	0·03
	GraRep	*mean rank*	676·07	944·67	849·28	813·40	828·20	840·75
		MAP	0·05	0·01	0·02	0·03	0·03	0·03
	HOPE	*mean rank*	-	749·61	776·90	803·42	838·26	874·51
		MAP	-	0·12	0·11	0·10	0·08	0·06
	LINE	*mean rank*	387·65	360·36	459·23	432·79	423·59	425·38
		MAP	0·07	0·06	0·06	0·05	0·06	0·08
	node2vec	*mean rank*	490·53	458·11	462·65	459·00	453·37	450·80
		MAP	0·02	0·03	0·04	0·04	0·04	0·04
	SDNE	*mean rank*	917·78	836·11	823·17	960·99	931·31	991·00
		MAP	0·04	0·10	0·10	0·02	0·04	0·02
Poincaré Embeddings	Pioncare Gensim	*mean rank*	765·08	734·58	747·20	750·99	739·38	745·25
		MAP	0·03	0·03	0·03	0·03	0·03	0·03
	Pioncare Cpp	*mean rank*	438·84	428·82	441·85	449·64	452·56	457·11
		MAP	0·06	0·09	0·09	0·09	0·09	0·09
	Pioncare Numpy	*mean rank*	935·95	880·16	861·51	874·15	892·52	879·84
		MAP	0·01	0·02	0·02	0·02	0·02	0·02
	Pioncare Pytorch	*mean rank*	1169·85	1151·57	1167·01	1164·53	1169·49	1165·13
		MAP	0·08	0·08	0·08	0·08	0·08	0·08
	Pioncare GloVe	*mean rank*	1268·48	1263·33	1250·31	1169·00	1165·30	1003·67
		MAP	0·01	0·01	0·01	0·03	0·04	0·06

Table 1: PhySH reconstruction from APS datasets, with word embeddings trained on raw text, graph embeddings and Poincaré embeddings trained on co-occurrence of PhySH terms in raw text. We only include each embedding's optimal result in the table.

- Word embeddings: CBOW and Skipgram (Mikolov et al., 2013), fastText (Joulin et al., 2017), GloVe(Pennington et al., 2014)[2].
- Graph embeddings: deepWalk (Perozzi et al., 2014), node2vec (Grover and Leskovec, 2016), LINE (Tang et al., 2015), LLE (Roweis and Saul, 2000), HOPE (Ou et al., 2016), GF(Ahmed et al., 2013), SDNE(Wang et al., 2016)[3].
- Poincaré embeddings: Poincaré-gensim[4], Poincaré-cpp [5], Poincaré-pytorch[6], Poincaré-numpy[7], Poincaré-glove[8] (Tifrea et al., 2019).

.

Word embeddings are trained on *title* and *abstract* of APS publications. The PhySH terms' embedding vectors will be extracted for taxonomy

[2]CBOW, Skipgram and fastText are trained by https://github.com/NIHOPA/word2vec_pipeline. GloVe is trained by https://github.com/stanfordnlp/GloVe

[3]Graph embeddings are implemented by OpenNE repository https://github.com/thunlp/OpenNE

[4]https://radimrehurek.com/gensim/models/poincare.html

[5]https://github.com/TatsuyaShirakawa/poincare-embedding.git

[6]https://github.com/facebookresearch/poincare-embeddings

[7]https://github.com/nishnik/poincare_embeddings.git

[8]https://github.com/alex-tifrea/poincare_glove

	Model Name	Metric	Dimensions					
			5	10	20	50	100	200
Graph Embeddings	deepWalk	*mean rank*	357·26	496·36	546·02	537·64	525·74	519·82
		MAP	0·22	0·19	0·21	0·22	0·22	0·23
	GF	*mean rank*	277·24	125·89	50·67	8·90	2·93	9·79
		MAP	0·10	0·35	0·58	0·65	0·66	0·66
	GraRep	*mean rank*	-	78·87	34·45	22·19	13·45	82·18
		MAP	-	0·49	0·53	0·56	0·58	0·57
	HOPE	*mean rank*	-	561·03	758·32	691·95	615·45	515·47
		MAP	-	0·64	0·47	0·43	0·43	0·45
	LINE	*mean rank*	489·49	344·14	141·35	34·32	15·84	10·00
		MAP	0·04	0·07	0·23	0·52	0·60	0·62
	node2vec	*mean rank*	265·65	264·94	265·69	264·81	269·52	265·20
		MAP	0·33	0·34	0·35	0·35	0·34	0·35
	SDNE	*mean rank*	72·58	33·18	478·27	517·55	512·10	492·46
		MAP	0·37	0·54	0·12	0·04	0·02	0·02
Poincaré Embeddings	Pioncare Gensim	*mean rank*	8·08	6·58	7·04	7·43	6·63	6·20
		MAP	0·61	0·61	0·62	0·61	0·62	0·61
	Pioncare Cpp	*mean rank*	12·04	11·74	8·12	6·75	8·17	6·95
		MAP	0·61	0·61	0·62	0·62	0·62	0·62
	Pioncare Numpy	*mean rank*	382·52	291·56	272·80	232·12	249·01	247·75
		MAP	0·46	0·53	0·56	0·58	0·58	0·59
	Pioncare Pytorch	*mean rank*	3·83	3·22	2·88	**2·61**	2·80	2·82
		MAP	0·93	**0·94**	**0·94**	**0·94**	**0·94**	**0·94**

Table 2: PhySH reconstruction from PhySH hypernym-hyponym pairs. Since there is no context infomation, word embeddings are not applicable here.

reconstruction. Graph embeddings and Poincaré embeddings are trained on the co-occurrence of PhySH terms in each of APS publications. As with (Nickel and Kiela, 2017), we also train graph embeddings and Poincaré embeddings on PhySH hypernym-hyponym pairs.

Taxonomy Reconstruction: We follow (Nickel and Kiela, 2017) to reconstruct taxonomy based on embedding vectors. For each embedding vector in Poincaré disk space, which is denoted as $\mathcal{B}^d = \{x \in R^d, \|x\|_2 \leq 1\}$. The norm of each vector can measure the radius of each vector, while the hyperbolic distance can measure the closeness of two vectors. The closest two are assigned as hypernym-hyponym pairs. The hyperbolic distance of two vector points $u, v \in \mathcal{B}^d$ is calculated follow as (Nickel and Kiela, 2017).

$$d_H(u,v) = arcosh\left(1 + 2 * \frac{\|u-v\|^2}{(1-\|u\|^2)(1-\|v\|^2)}\right) \quad (1)$$

The distance could only tell how semantically close are the node pairs (u, v). But which one is the parent node is not answered. One property that makes hyperbolic space outstanding for the hierarchical structure is that the hyperbolic disc area and circle length grow exponentially with its radius. Node with the smaller norm is the higher-level term.

4 Evaluation and Results

4.1 Evaluation Datasets

APS (American Physical Society) has made available their publications data for researchers with the total number of $661,209$ articles and citations and dates back to 1893[9]. We utilize the article metadata datasets. PhySH (Physics Subject Headings) is the Physics concept hierarchy. It is used to organize publications of APS. It is open-source on Github[10]. APS metadata datasets only contain *title* field, we retrieve *abstract* from Web of Sciences database[11].

4.2 Evaluation Metrics

mean rank and *MAP* metrics are used to measure taxonomy reconstruction performance. *mean rank* is calculated for each node's distance of ground truth children against all other nodes. *MAP* is the mean average precision at the threshold of each correctly retrieved child.

$$\text{mean\_rank}(u) = \frac{sp(u)}{sp(u) + lp(u)} \in [0, 1] \quad (2)$$

$lp(u)$ is the furthest length from node u to its descendants. $sp(u)$ is the shortest length from node u to root node. The optimal embedding should score a low *mean rank* and a high *MAP*.

4.3 PhySH Reconstruction Evaluation

PhySH Reconstruction From Raw Text. In this experiment, we extract co-occurrence of PhySH terms in each APS publication. The graph embeddings are trained on the co-occurrence graph. Poincaré embeddings are trained on the noisy co-occurrence pairs. Word embeddings are trained on APS publication raw texts. PhySH terms' representation vectors are extracted from word vectors in the postprocessing step.

Table 1 is the performance of PhySH reconstruction by learning representation from raw APS datasets[12]. None of the embeddings get the best result in both metrics. Word embeddings like CBOW achieve better *MAP*, while graph embeddings like deepWalk outperform in *mean rank*. Poincaré embeddings did not show any superior. Learn PhySH

[9] https://journals.aps.org/datasets

[10] https://github.com/physh-org/PhySH

[11] Web of Science is a commercial database of Clarivate Analytics, it can be accessed by most universities and institutions

[12] We experiment each embedding with different hyperparameters by grid search, we present the optimal performance of each embedding in the tables.

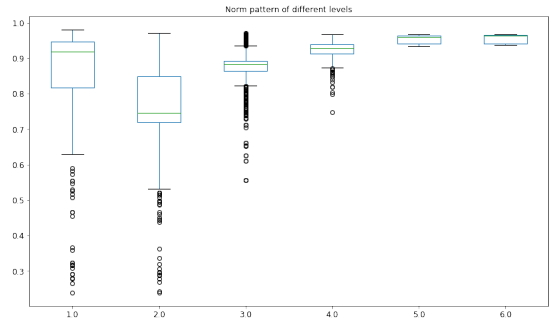


Figure 2: Norm of Poincaré embedding vector at different taxonomy levels

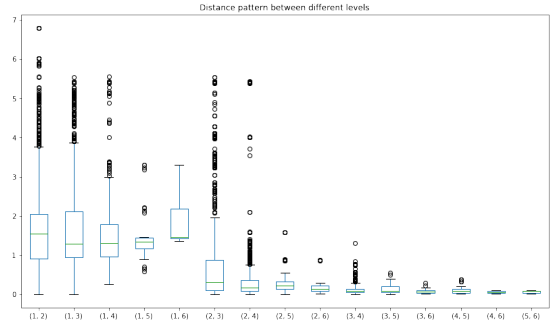


Figure 3: Distance of Poincaré embedding vector across different taxonomy levels

from noisy co-occurrence pairs are much more complicated than the mammal tree of the WordNet described in the origin paper(Nickel and Kiela, 2017). We can conclude Poincaré embeddings are not sufficient for learning and representing from the co-occurrence pairs.

PhySH Reconstruction From PhySH. Table 2 is the performance of PhySH reconstruction by learning representation from PhySH hypernym-hyponym pairs. The graph embeddings are trained on the PhySH hypernym-hyponym graph. Poincaré embeddings are trained on the PhySH hypernym-hyponym pairs.

In this experiment, Poincaré's official implementation *Poincaré-Pytorch* wins with far better results than other algorithms. This is because Poincaré is trained with the loss function designed to learn hierarchies, while graph embeddings are trained to learn from neighbors and global graph structure. However, *GF* at dimension 100 and *LINE* at dimension 200 also get very good performance.

4.4 The Hierarchical Characteristics of PhySH Poincaré embedding

If we understand the successful representation characteristics of taxonomy hierarchical relations, it will be the help of taxonomy construction. We will analyze what are the hierarchical characteristics of PhySH preserved by Poincaré embeddings in this section.

In Figure 2, we visualize how the norm value varies in different PhySH level. There is a clear pattern from taxonomy level 2 to level 6: lower-level terms have bigger norm values. It means lower terms are further from the root term. The pace of the decrease of the norm in lower levels seems to decelerate, which needs to be further validated. However, the norm of level 1 terms is rather distributed, which we think is the points where Poincaré embedding fails.

In Figure 3, we compare the distance of terms over different PhySH levels. The ancestor nodes are further than parent nodes. For each node, its distance to the child is smaller than the distance to parent, and the distance to the child is nearly half as the distance to parent. These patterns are important for a successful representation of taxonomy.

5 Conclusion and Future Work

we compare word embeddings, graph embeddings, and Poincaré embeddings by reconstructing PhySH. We consider two scenario case: reconstructing from raw texts and reconstructing from existing PhySH. The experiment shows even though Poincaré embeddings far outweigh other embeddings in reconstructing PhySH from PhySH, it is also not competent as other embeddings in reconstructing PhySH from raw APS texts.

We further demystify what is the success of Poincaré embeddings in reconstructing PhySH from PhySH. The future work would be how to design a powerful taxonomy induction algorithm which could benifit from the characteristics of our paper.

Acknowledgments

The authors thank anonymous reviewers for their helpful comments that improved the manuscript. The authors also thank American Physical Society for providing access to the APS Datasets.

This work is supported in part by the National Natural Science Foundation of China under contracts No 71950003, and the project "Design and Research on a Next Generation of Open Knowledge Services System and Key Technologies" (2019XM55).

6 References

References

Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J. Smola. 2013. Distributed large-scale natural graph factorization. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 37–48, New York, NY, USA. ACM.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka, Jr., and Tom M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 101–110, New York, NY, USA. ACM.

Christopher De Sa, Albert Gu, Christopher RÃł, and Frederic Sala. 2018. Representation tradeoffs for hyperbolic embeddings. *Proceedings of machine learning research*, 80.

Bhuwan Dhingra, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl. 2018. Embedding text in hyperbolic spaces. In *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, pages 59–69, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Zornitsa Kozareva and Eduard Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1110–1118, Cambridge, MA. Association for Computational Linguistics.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, pages 1048–1056, Columbus, Ohio. Association for Computational Linguistics.

Juan J. Lastra-DÃ■az, Josu Goikoetxea, Mohamed Ali Hadj Taieb, Ana GarcÃ■a-Serrano, Mohamed Ben Aouicha, and Eneko Agirre. 2019. A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art. *Engineering Applications of Artificial Intelligence*, 85:645 – 665.

Marc T Law, Jake Snell, and Richard S Zemel. 2019. Lorentzian distance learning.

Matthias Leimeister and Benjamin J. Wilson. 2019. Skip-gram word embeddings in hyperbolic space.

Anh Tuan Luu, Jung-jae Kim, and See Kiong Ng. 2014. Taxonomy construction using syntactic contextual evidence. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 810–819, Doha, Qatar. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Roberto Navigli and Paola Velardi. 2004. Learning domain ontologies from document warehouses and dedicated web sites. *Comput. Linguist.*, 30(2):151–179.

Kim Anh Nguyen, Maximilian KÃűeper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical Embeddings for Hypernymy Detection and Directionality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 233–243, Copenhagen, Denmark.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6341–6350. Curran Associates, Inc.

Farhad Nooralahzadeh, Lilja Øvrelid, and Jan Tore Lønning. 2019. Evaluation of domain-specific word embeddings using knowledge resources. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pages 1438–1445.

Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114. ACM.

Sebastian Padó and Mirella Lapata. 2003. Constructing semantic space models from parsed corpora. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Sapporo, Japan. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710, New York, NY, USA. ACM.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Ivan Sanchez and Sebastian Riedel. 2017. How well can we predict hypernyms from word embeddings? a dataset-centric analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 401–407, Valencia, Spain. Association for Computational Linguistics.

Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T. Vanni, Brian M. Sadler, and Jiawei Han. 2018. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 2180–2189, New York, NY, USA. ACM.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*. ACM.

Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. 2019. Poincare glove: Hyperbolic word embeddings. In *International Conference on Learning Representations*.

Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2017. A Short Survey on Taxonomy Learning from Text Corpora: Issues, Resources and Recent Advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1992, pages 1190–1203, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1225–1234, New York, NY, USA. ACM.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 481–492, New York, NY, USA. ACM.

Ichiro Yamada, Kentaro Torisawa, Jun'ichi Kazama, Kow Kuroda, Masaki Murata, Stijn De Saeger, Francis Bond, and Asuka Sumida. 2009. Hypernym discovery based on distributional similarity and hierarchical structures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 929–937, Singapore. Association for Computational Linguistics.

Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. 2015. Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS '15, pages 12:1–12:8, New York, NY, USA. ACM.

Combining Representations For Effective Citation Classification

Claudio Moisés Valiense de Andrade
Federal University of Minas Gerais
claudio.valiense@dcc.ufmg.br

Marcos André Gonçalves
Federal University of Minas Gerais
mgoncalv@dcc.ufmg.br

Abstract

We describe our participation in two tasks organized by WOSP 2020, consisting of classifying the context of a citation (e.g., background, motivational, extension) and whether a citation is influential in the work (or not). Classifying the context of an article citation or its influence/importance in an automated way presents a challenge for machine learning algorithms due to the shortage of information and inherently ambiguity of the task. Its solution, on the other hand, may allow enhanced bibliometric studies. Several text representations have already been proposed in the literature, but their combination has been underexploited in the two tasks described above. Our solution relies exactly on combining different, potentially complementary, text representations in order to enhance the final obtained results. We evaluate the combination of various strategies for text representation, achieving the best results with a combination of TF-IDF (capturing statistical information), LDA (capturing topical information) and Glove word embeddings (capturing contextual information) for the task of classifying the context of the citation. Our solution ranked **first** in the task of classifying the citation context and *third* in classifying its influence.

1 Introduction

Data science is becoming more and more popular with its largest data community being *Kaggle*[1], a platform that hosts several data mining and machine learning tasks and challenges. In 2020, the 8th International Workshop on Mining Scientific Publication (WOSP)[2], through Kaggle, promoted two challenges consisting of: 1) classifying the context of a citation in one of the six existing classes (e.g., background, motivational, extension) and 2) a binary task where the goal is to identify the importance of a citation for a given work. The competition overview was presented by Kunnath et al. (2020) (N. Kunnath, 2020).

An example of the citation context taken from the dataset is:*"In the future we are planning to experiment with different ways of calculating relatedness of the sequences to the descriptions, such as with computing similarity of embeddings created from the text fragments using approaches like Doc2Vec (#AUTHOR_TAG and Mikolov, 2014)"*, where #AUTHOR_TAG tag means the quote being classified. In this case, for the context classification challenge, this citation belongs to the Future class, and for the influence classification task, this citation is considered influential.

For the sake of the defined classification tasks, the citation text can be represented in several ways (e.g. TF-IDF, word embeddings, text graph), each representation capturing or focusing on a different aspect of the task. For instance, the traditional TF-IDF representation captures statistical aspects of the text and the specificity of certain words in the collection (IDF component). Topic modeling strategies such as LDA identify patterns of recurrent topics (i.e., clusters of words) in the text. Word embeddings are vectorial representations of words, sentences and whole documents aimed at capturing word co-occurrence patterns and contextual information.

Our main hypothesis here is that these different sources of information are somewhat complementary and, when combined, have the potential to enhance classification effectiveness. By exploring such ideas, we were able to reached the **first** place in the multiclass classification task promoted by *WOSP* and *third* in the binary influence classification task, with further improvements after the closing of the challenge, as we shall see.

In the competition there are two types of scores:

[1] http://www.kaggle.com
[2] https://wosp.core.ac.uk/jcdl2020

1) the public score, calculated based on 50% of the test data and 2) the private score, based on the results of the other 50% and only displayed after the closing of the challenge. Using a combination of TF-IDF, LDA and Glove embedding representations, as aforementioned, along with a Passive-Aggressive classifier (Crammer et al., 2006), our final result improved by 3.62% the scores of the best isolated representation in the public score, while for the private score this difference was up 6.91%. For the task of assessing the importance of the citation, we obtained an improvement of 1.39% for the public score and 3.07% for the private one, by adding a feature that identifies that the author of the quote is the same author who is making the citation.

To guarantee the reproducibility of our solution, all the code is available on github[3][4] and we create an image through the docker with the entire process configured.

2 Related Work

Some papers addressed the problem of classifying the citation for context and influence. Jurgens et al. (2018) (Jurgens et al., 2018) aimed at classifying the context of a citation, similarly to the challenge, into six possible classes. To train the model, they used structural information from the text, lexical, grammatical, etc. In that work, specific terms have higher weights in relation to the others, for example, "we extend" have more importance in the classification of the context "Extends". Unlike their work, we exploit only information from titles and the context of the citation with combinations of representations.

In Valenzuela et al. (2015) (Valenzuela et al., 2015), if the citation is connected to related works or is used for comparison between methods, the author considers it as an incidental citation. If the cited method is used in the work or the current work is an extension of the cited one, the citation is considered important. These are the two possibilities in the challenge of the influential classification competition. In that work, the authors used 12 features, among them, the citation count per section, similarity between abstracts, etc.

Pride et al. (2017) (Pride and Knoth, 2017) also dealt with the binary task of classifying an influential citation. In that work, the authors expose the problem of extracting data from *pdf* when there is no structured data. Their work analyzed the features of previous works and the impacts of adding specific features to the model. Unlike Valenzuela and Pride's works, we do not use any information other than the titles and context of the citation, data provided through the Kaggle.

3 Methodology

In this section we present our methodology, consisting of of applying preprocessing to the data, creating different representations to explore the importance of individual words (TF-IDF), group similar concepts shared by terms (LDA) and explore the semantics and context of terms (Glove). After this step, we join (concatenate) the representations to train the classifier, aiming to predict the class of the test set.

The dataset contains eight fields, for each document we combine fields Citing Paper Title, Cited Paper Title and Citation Context (hereafter called *citation text* or simply citation (Cit)) on a single line, separated by space. We use this file as input to the data preprocessing algorithm, which consists of: 1) turning the text into lowercase, 2) removing accents from words and 3) applying a parser that replaces specific strings with tags fixes, for example, number by "parserNumber".

After preprocessing the citation text, we initially use the TF-IDF (Luhn, 1957) representation. It exploits both, unigrams and bigrams. where the bigram is the combination of the current term and the next one. In more details, TF quantifies the frequency of the terms (unigrams or bigrams) and the IDF measures their inverse frequency of document, giving higher weights to terms that occur less frequently in documents (higher discriminative power).

Another representation we use is the Latent Dirichlet Allocation (LDA) with online variational Bayes algorithm (LDA) (Hoffman et al., 2010). It groups terms into similar concepts called topics. Topics are learned as a probability distribution on the words that occur in each document, using as input the original TF-IDF representation. For each document, a score is associated with each topic and a citations may be seen as a combination of topics with different likelihoods.

We tested some values for the hyperparameter that defines the number of topics in the dataset.

[3] https://github.com/claudiovaliense/wosp_2020_3c-shared-task-purpose

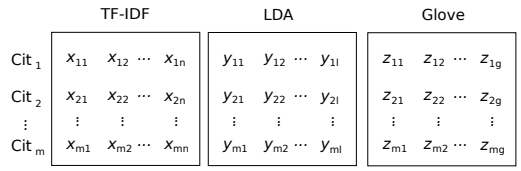
[4] https://github.com/claudiovaliense/wosp_2020_3c-shared-task-influence

We tested 6, 10, 50 and 100 topics. The one that produced the best results in the training data was 6 topics. That is, the algorithm creates 6 groups of words in the training and at the moment it receives a new test citation Cit_i, it assigns a score (likelihood) to each of the groups for Cit_i.

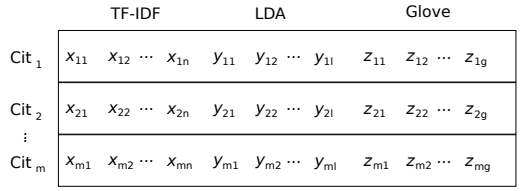
The last exploited representation was Glove embeddings (Pennington et al., 2014), an unsupervised learning algorithm to obtain vector representations for words. For each term *t* present in a citation *cit*, we average (pool) their respective Glove vectors to create a representation for the whole citation.

Each textual representation generates a number of features for each citation. In TF-IDF, this number corresponds to the amount of unique terms existent in the entire dataset. For LDA, this is the number of defined topics, in our case 6. For the Glove embeddings we use the vector representation with 300 dimensions, thus generating 300 features for each citation. Figure 1a presents the features for each citation cit_i, where m represents the number of citations and n, l and g the number of features generated by each representation.

Finally, to combine the representations, we use a simple concatenation of all available representations (Feature Union[5]) resulting in a single vector, as shown in Figure 1b.



(a) Representations.



(b) Feature Union.

Figure 1: Combining Representations.

4 Experiment

The dataset was created based on the methodology developed by Pride et al. (2019) (Pride et al., 2019). By means of an automatic system, authors (or external evaluators) can select to which group the citations are most related to (context) and whether the citation is described as central to the work.

Based on this methodology, 3000 citations with defined classes were made available through the Kaggle for training along with 1000 test citations that should be classified according to the defined classes. Table 1 describes the training data, with: Number of citations (|D|), Median of the amount of term (M(T)), Number of classes (|C|), Number of citations of the largest (Max(C)) and smallest (Min(C)) class. Note that the dataset is very unbalanced – the largest class has 1648 citations while the smallest contains only 62.

Table 1: Dataset Metadata.

Name	|D|	M(T)	|C|	Max(C)	Min(C)
Context	3000	55.00	6	1648	62
Influence	3000	55.00	2	1568	1432

4.1 Classification and Parameter Tuning

Among the classifiers we tested, Passive Aggressive, Stochastic Gradient Descent (SGD) and Linear SVM presented the best results in preliminary experiments and were selected to be used in the challenge. For each of them we optimized the hyperparameter through stratified cross-validation (10 folds) in the training set. For the Passive-Aggressive and Linear SVM classifiers, we evaluated the *C* parameter varying among $[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}]$ and for SGD we vary the *alpha* parameter among $[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$. We used a potency of 10 to avoid overfitting the classifier.

Table 2 shows the result of the process of tuning the parameters with the cross-validation procedure. We present the parameters that obtained the highest scores in cross-validation, the macro F1 score, in parentheses, the standard deviation and the time in seconds spent by each classifier. As can be seen, Linear SVM is about 5 times slower than Passive-Aggressive, while there is a statistical tie in the final (training) result. Since we would need to test many alternatives and configurations in our trials, we decided to choose Passive-Aggressive as the final classifier for the challenge. The Passive-Aggressive algorithms are a family of algorithms for large-scale learning not requiring a learning rate (Crammer et al., 2006). However, contrary to the Perceptron, they include a regularization parameter C.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.FeatureUnion.html

Table 2: Result of classifiers in the evaluation.

Classifier	Best Parameter	Macro-F1	Time (s)
PA	C [10^0]	0.1846 (0.044)	94.36
SGD	alpha [10^{-4}]	0.1740 (0.033)	21.38
SVM	C [10^0]	0.1953 (0.049)	486.75

Table 3: Kaggle Score

Method	Public	Private	Task
tfidf	0.19829	0.19425	Context
lda	0.12923	0.15826	Context
glove	0.12047	0.11489	Context
tfidf+lda	0.19124	0.19572	Context
tfidf+glove	0.19945	0.20037	Context
tfidf+lda+glove	**0.20548**	**0.20560**	**Context**
tfidf	0.59108	0.54747	Influence
lda	0.30458	0.32249	Influence
glove	0.30458	0.32249	Influence
tfidf+lda	0.32707	0.36156	Influence
tfidf+glove	0.30458	0.32249	Influence
tfidf+lda+glove	0.30458	0.32249	Influence
tfidf+same_author	**0.59932**	**0.56431**	**Influence**

4.2 Result

In the Kaggle challenge, the evaluation was based on Macro-F1, probably due to the high skewness of the Context task. For each new submission the score is calculated based on 50% of the test data (public score). The results of the other 50% (private score) were only displayed at the closing of the challenge. The final result of the competition was based on the private score. Table 3 presents the results of the individual representations as well as their combinations, considering the public and private scores.

For the classification of topics, the strategy that presented the best results used the combination of the three aforementioned representations – TF-IDF, LDA and GLOVE. Notice that in this task, the performance of TF-IDF is already high, better than LDA and Glove.

In the classification of influential citations, TF-IDF alone produces the best results. Combinations of representations using LDA, Glove or both, showed a reduction in the final score. In this task, the effectiveness of both LDA and Glove are far from TF-IDF, about 50% less effective. We hypothesize that the concatenation of the representations produce a very high dimensional space that, along with the not so good performance of LDA and Glove, exacerbates issues of noise and overfitting in this binary task. We will further investigate this in future work. After the submission deadline, we added a feature that captured whether the author being cited is the same author of the article that quotes, this feature has improved the final result (tfidf+same_author).

We should stress that the excellent performance of TF-IDF alone is consistent with recent results that show that TF-IDF, when coupled with a strong, properly tuned classifier, is still one of the best text representations, better than certain word embeddings for classification tasks, (Cunha et al., 2020).

5 Conclusion

In this paper we described our participation in the citation classification tasks organized by WOSP 2020. We focused on evaluating combinations of textual representations – statistical information with TF-IDF, topical with LDA, and contextual and co-occurence information with Glove word embeddings – and the impact of each one on the final result. Our solution relied on exploring multiple, potentially complementary, representations to add their benefits as they potentially capture different textual aspects. We use the Passive-aggressive classifier, the best and faster in a preliminary evaluation for the task, optimizing its hyperparameters through stratified folded cross validation within the training set. TF-IDF demonstrated to be a very powerful representation when used with a strong, properly tuned classifier, confirming recent results that it may better than certain alternatives (e.g., embeddings) for specific tasks (Cunha et al., 2020). But its combination with other representations indeed did help to improve results, as initially hypothesized. Overall, our solution achieved very good results, reaching the **first** place in the task of classifying the context of a citation and *third* in the classification of influential citations (with post-deadline improvements).

As future work, we intend to evaluate combinations with new representations, e.g., MetaFeatures (Canuto et al., 2018; Canuto et al., 2016, 2019) and Cluwords (Viegas et al., 2018, 2019, 2020). Due to the shortage of information, enhancing citation data with automatic tagging information (Belém et al., 2019, 2014, 2011) seems as a promising strategy to obtain more data.

Acknowledgments

This work was partially supported by CNPq, Capes and Fapemig.

References

Fabiano Belém, Eder Ferreira Martins, Tatiana Pontes, Jussara M. Almeida, and Marcos André Gonçalves. 2011. Associative tag recommendation exploiting multiple textual features. In *Proceeding of the 34th ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 1033–1042.

Fabiano Muniz Belém, André G. Heringer, Jussara M. Almeida, and Marcos André Gonçalves. 2019. Exploiting syntactic and neighbourhood attributes to address cold start in tag recommendation. *Inf. Process. Manag.*, 56(3):771–790.

Fabiano Muniz Belém, Eder Ferreira Martins, Jussara M. Almeida, and Marcos André Gonçalves. 2014. Personalized and object-centered tag recommendation methods for web 2.0 applications. *Inf. Process. Manag.*, 50(4):524–553.

S. Canuto, D. X. Sousa, M. A. Gonçalves, and T. C. Rosa. 2018. A thorough evaluation of distance-based meta-features for automated text classification. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2242–2256.

Sérgio D. Canuto, Marcos André Gonçalves, and Fabrício Benevenuto. 2016. Exploiting new sentiment-based meta-level features for effective sentiment analysis. In *Proc. of the 9th ACM Conf. on Web Search and Data Mining*, pages 53–62.

Sérgio D. Canuto, Thiago Salles, Thierson Couto Rosa, and Marcos André Gonçalves. 2019. Similarity-based synthetic document representations for meta-feature generation in text classification. In *Proc. of the 42nd ACM SIGIR Conf. on Research and Development in Information Retrieval, SIGIR 2019*, pages 355–364.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585.

Washington Cunha, Sérgio Canuto, Felipe Viegas, Thiago Salles, Christian Gomes, Vitor Mangaravite, Elaine Resende, Thierson Rosa, Marcos André Gonçalves, and Leonardo Rocha. 2020. Extended pre-processing pipeline for text classification: On the role of meta-feature representations, sparsification and selective sampling. *Information Processing & Management (IP&M)*, 57(4).

Matthew D. Hoffman, David M. Blei, and Francis Bach. 2010. Online learning for latent dirichlet allocation. In *Proc. of the 23rd Conf. on Neural Information Processing Systems - Volume 1*, page 856–864.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

H. P. Luhn. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.*, 1(4):309–317.

David; Gyawali Bikash; Knoth Petr N. Kunnath, Suchetha; Pride. 2020. Overview of the 2020 wosp 3c citation context classificationtask. in: Proceedings of the 8th international workshop on mining scientific publications. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL 2020)*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

David Pride, Petr Knoth, and Jozef Harag. 2019. ACT: an annotation platform for citation typing at scale. In *19th ACM/IEEE Joint Conf. on Digital Libraries, JCDL 2019*, pages 329–330. IEEE.

David T. Pride and Petr Knoth. 2017. Incidental or influential? - a decade of using text-mining for citation function classification. In *ISSI*.

Marco Valenzuela, Vu Ha, and Oren Etzioni. 2015. Identifying meaningful citations. In *AAAI Workshop: Scholarly Big Data*, volume WS-15-13 of *AAAI Workshops*.

Felipe Viegas, Sérgio Canuto, Christian Gomes, Washington Luiz, Thierson Rosa, Sabir Ribas, Leonardo Rocha, and Marcos André Gonçalves. 2019. Cluwords: Exploiting semantic word clustering representation for enhanced topic modeling. In *Proc. of the 12th ACM Conf. on Web Search and Data Mining*, page 753–761.

Felipe Viegas, Washington Cunha, Christian Gomes, Antônio Pereira, Leonardo C. da Rocha, and Marcos André Gonçalves. 2020. Cluhtm - semantic hierarchical topic modeling based on cluwords. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online*, pages 8138–8150.

Felipe Viegas, Washington Luiz, Christian Gomes, Amir Khatibi, Sérgio D. Canuto, Fernando Mourão, Thiago Salles, Leonardo C. da Rocha, and Marcos André Gonçalves. 2018. Semantically-enhanced topic modeling. In *Proc. of the 27th ACM Conf. on Information and Knowledge Management, CIKM 2018*, pages 893–902.

Scubed at 3C task A - A simple baseline for citation context purpose classification

Shubhanshu Mishra
shubhanshu.com
mishra@shubhanshu.com

Sudhanshu Mishra
Indian Institute of Technology Kanpur
Kanpur
India
sdhanshu@iitk.ac.in

Abstract

We present our team Scubed's approach in the 3C Citation Context Classification Task, Subtask A, citation context purpose classification. Our approach relies on text based features transformed via tf-idf features followed by training a variety of models which are capable of capturing non-linear features. Our best model on the leaderboard is a multi-layer perceptron which also performs best during our rerun. Our submission code for replicating experiments is at: https://github.com/napsternxg/Citation_Context_Classification.

1 Introduction

The number of research papers has increased exponentially in recent years. In order to efficiently access this scientific resource, we need automated solutions for extracting information from these records. Citations in research papers are important for multiple reasons e.g. comparing novelty (Mishra and Torvik, 2016), expertise (Mishra et al., 2018a), and self-citation patterns (Mishra et al., 2018b). For people new to the field, they are an important resource to increase their knowledge whereas for experts in the field they act as useful pointers to summarize the paper. Citations are also used to measure various indexes which showcase the influence and reach of the researchers in their field. However, these indexes give equal weight to each citation. It has been established that all citations are not equal (N. Kunnath et al., 2020; Mishra et al., 2018b). In many cases, cited papers are used as examples. Often, they are not influential to the paper itself.

In this paper we describe our team, Scubed's entry for the citation context purpose classification shared task (N. Kunnath et al., 2020). This work aims to develop models that can identify the purpose of citations in the research papers, and hence can then be used to produce better indexes and make research more easily accessible to everyone.

1.1 Related Work

There has been a significant amount of work done in this area to better understand the significance of citations in a paper (N. Kunnath et al., 2020). As the number of research papers increase with time, the algorithms for suggesting research papers become more and more important. These algorithms are a deciding factor for lots of measures of a researcher's influence in a field. The no. of citations of a paper are important for deciding measures such as h-index (Hirsch, 2005) and g-index (Egghe, 2006). These are influential measures for describing the significance of a researcher in a field. Scholars have argued that all of the citations in a paper should not have the same weight while determining the impact and reach of a paper. Moras et. al (Moravcsik and Murugesan, 1975) showed, that many references in research papers are redundant and quite often share little context with the citing paper. There have been many techniques for classifying citations as influential. However, one of the strongest baseline for this task is the prior citation count of the cited paper. Works of (Chubin and Moitra, 1975) show the effectiveness of citation count in determining influence. The work of (Zhu et al., 2015) points out suitable features for this task. They evaluated the performance of 5 classes of features, count, position, similarity, context and miscellaneous. They determined that counting the number of times a citation is referenced in a paper is the best estimator to determine the influence of a citation. (Hou et al., 2011) also showed that the count of a citation in a research paper is a simple and effective technique to assign its scientific contribution and influence. (Nazir et al., 2020) applied SVM, Random Forests and Kernel Linear Regression classifiers to identify important

and non-important citations. They used citation count and similarity scores using tf-idf features to train their models. Their results show that these techniques produce an improved precision score of 0.84 in these tasks.

2 Task and Data Description

This paper focuses on the WOSP 3C shared sub-task B. In this sub-task, we were required to classify the citation context in research papers on the basis of their influence and purpose in the paper. For this shared task we used the ACL-ARC dataset(Jurgens et al., 2018) . The dataset consisted of 3000 labeled data-points annotated using the ACT platform (Pride et al., 2019). The data provided contains the following fields:

- Unique Identifier
- COREID of Citing Paper
- Citing Paper Title
- Citing Paper Author
- Cited Paper Title
- Cited Paper Author
- Citation Context
- Citation Class Label
- Citation Influence Label

To identify the citation being considered a #AUTHORTAG is placed in the citation. For this task the Citation Class Label field was ignored. This was a multi-label classification task, where the following target labels were used :

- **BACKGROUND**
- **COMPARES_CONTRASTS**
- **EXTENSION**
- **FUTURE**
- **MOTIVATION**
- **USES**

To evaluate the models the macro-F1 score was used on the test data. The final score that was used to rank was not the public score but a different subset of data that was not visible to the participating teams. The teams were advised to make submissions that would perform the best overall and not just on the public subset.

3 Methodology

We utilize a simple approach based on text classification baseline methods. For the original submission we utilized a limited set of models. However, we trained additional models to conduct exhaustive evaluation for this paper. Below, we describe our workflow for pre-processing, feature extraction, and model-training.

3.1 Pre-Processing and Feature Extraction

The data provided was in raw text format which is not suitable for making predictions directly. In order to make useful predictions, it has to be first converted into numerical vector form that our models can process. The raw data consisted of columns having different attributes for which different feature extraction techniques had to be applied. For example, the *citing* and *cited title* consisted of a titles of the research papers whereas the *citation context* consisted of a description of the citation context. In order to efficiently process each column separately we used the *ColumnTransformer* module from the scikit-learn library (Pedregosa et al., 2011). Each of the column contained text data. To extract useful features from this text data we used the *TfidfVectorizer* from the scikit-learn (Pedregosa et al., 2011) library on each column. This generates the term frequency inverse document frequency(*tf-idf*) score for each of the texts in each column. The tf-idf score is a normalized count for the words occurring in the corpus. This type of feature however does not account for the position and inter-dependence of words. The tf-idf score is calculated as follows:

$$tf - idf(t, d) = tf(t, d) * idf(t) \tag{1}$$

$$idf(t) = \log\left(\frac{1+n}{1+df(t)}\right) + 1 \tag{2}$$

In the above equations, *tf* stands for term frequency which refers to the number of times a term *t* occurs in a document *d*. The *n* in **(2)** refers to the total number of documents present in the document set. *(Df(t))* refers to the document frequency which calculates the number of documents in the document set that contain the term *t*. The tf-idf score is a better feature compared to the count of words in a sentence. The tf-idf score down weights uninformative words like pronouns compared to more rare but informative words present in the document.

In the end we ended up using two version of text features for our models:

1. **Citing Context only (v1)**: uses only features extracted from *citation context* column. Our

hypothesis here is that citation context should have the highest signal for identifying how the citation is used.

2. **All features (v2)**: uses features extracted from *citation context* as well as *citing* and *cited title* column. Our hypothesis here is that using the combination of features from both citing and cited paper should improve the signal for identifying how the citation is used. However, we are also aware that this may also increase the proportion of noisy features.

3.2 Prediction Models

For this shared task we were allowed to submit a maximum of 5 models for evaluation on test data [1]. Our goal was to investigate usage of the most simple models based on proven linear and non-linear models which are faster and easier to train and deploy compared to the recent more powerful but resource hungry deep learning models. The following models were submitted for evaluation:

- **Logistic Regression Classifier (LR)**: A simple logistic regression model trained on the tf-idf features of 3 columns.
- **Random Forest (RF)**: Random Forest model with 100 trees in the forest and boot-strapping trained on the tf-idf features.
- **Gradient Boosting Classifier (GBT)**: A gradient boosted classifier with 100 boosting stages trained on the tf-idf features.
- **Multi-layer Perceptron Classifier (MLP)**: A 1 hidden layer multi-layer perceptron classifier with 100 nodes and Relu activation, optimized using Adam optimizer with a learning rate of 0.001 and momentum of 0.99.
- **Multi-layer Perceptron Classifier (MLP-3)**: A 3 hidden layer multi-layer perceptron classifier with 256, 256, and 128 nodes in the first, second and third layers with Relu activation optimized using Adam optimizer with a learning rate of 0.001 and momentum of 0.99.

All the models were trained using the scikit-learn library.

4 Results

Table 1 shows the the public and private leader board scores for each of our submissions for this task. Our MLP (v2) model performed best on the leader-board while similar to the top performing model (within 0.02 F1 score).

Table 1: Results for the Purpose Sub-task. 4* implies that according to the leader board our entry is better than the 4th position entry. The non-highlighted rankings are made on the basis of the leader board private scores visible to us.

S.No	Model	Private	Public	Rank
1	GBT	0.144	0.150	4*
2	RF2	0.144	0.142	4*
3	MLPC	0.182	0.176	3
6	**Best**	**0.206**	-	**1**

4.1 Replication model performance after leader board submission

After the final leader board ranking, we decided to replicate the model performance on the actual test set provided to us by the shared task organizers. Our evaluation scores may not match with the submitted solutions as the model changes on each run and we did not record the random seed for the original submission. This analysis was conducted to generate comparable results for all models across the training and test sets (see table 2), and to further inspect the performance of the model on each label (see table 3).

First, table 2 shows the evaluation scores of all the models on the test set. One consistent pattern emerges, v1 models which use only the citation context text as its feature, consistently perform much better than v2 models. Next, the best v1 as well as v2 models are MLP and MLP-3. It appears that inclusion of extra features leads to over-fitting which is also evident from the training evaluation scores.

Table 2: Model evaluation scores on the test data on retraining models after leader board ranking.

model	v1		v2	
	test	train	test	train
lr	0.135	0.296	0.120	0.281
rf	0.140	0.954	0.136	0.958
gbt	0.151	0.719	0.148	0.770
mlp-3	0.186	0.995	0.177	1.000
mlp	**0.187**	0.995	0.185	1.000

Second, in table 3 we investigate the per label

[1] https://www.kaggle.com/c/3c-shared-task-influence/rules

evaluation (in terms of F1 score) for each of the models. For both v1 and v2 features almost all models show similar performance on all labels. All models perform best on the Background label which is also the most frequent label. Overall, it appears that these baseline models are quite good at learning this task compared to other submissions, while being fast and easy to implement.

5 Discussion

Our results show that tradition tf-idf features give good performance for this shared task resulting in a strong baseline to compare against. Simple machine learning models like logistic regression, random forests, and gradient boosted trees perform well for this task but are superseeded by multi-layer prerceptron models. Furthermore, the citation context contains the maximum signal for predicting citation usage. We were able to achieve one of the top performances in the task within the number of submissions required in the task. Due to the small dataset, multiple submissions increase the likelihood of the models to over-fit to the test set. Furthermore, our methods show that deep learning methods (e.g. mlp and mlp-3) do give significant advantage over simpler machine learning methods. The minor loss in performance is acceptable compared to the increased speed and low computation of simple machine learning models.

Further analysis reveals that MLP based models are indeed over-fitting to the training data as shown by near perfect F1-score on the training data (see 2). Additionally, GBT models consistently achieve much better performance on the test set compared to other models, including RF model which was our best entry on the leader board. Furthermore, the highest performing label is the Influential label. All models (except LR) perform the worse on the Incidental when using all text features but when only using citation context, the label performance is similar across labels.

6 Conclusion

Our team 'Scubed' submitted 3 models for the citation context classification based on purpose task. Out of the submitted models the multi-layer perceptron classifier performed the best on the test set achieving third position in this task. This model gave a private score of **0.18146** on the test set. We were able to achieve competitive results under minimum trials using fast and computationally cheap machine learning models.

References

Daryl E. Chubin and Soumyo D. Moitra. 1975. Content analysis of references: Adjunct or alternative to citation counting? *Social Studies of Science*, 5(4):423–441.

Leo Egghe. 2006. Theory and practise of the g-index. *Scientometrics*, 69(1):131–152.

Jorge Hirsch. 2005. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102:16569–72.

Wen-Ru Hou, Ming Li, and Deng-Ke Niu. 2011. Counting citations in texts rather than reference lists to improve the accuracy of assessing scientific contribution. *BioEssays*, 33(10):724–727.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association of Computational Linguistics*.

Shubhanshu Mishra, Brent D. Fegley, Jana Diesner, and Vetle I. Torvik. 2018a. Expertise as an aspect of author contributions. In *WORKSHOP ON INFORMETRIC AND SCIENTOMETRIC RESEARCH (SIG/MET)*, Vancouver.

Shubhanshu Mishra, Brent D. Fegley, Jana Diesner, and Vetle I. Torvik. 2018b. Self-citation is the hallmark of productive authors, of any gender. *PLOS ONE*, 13(9):e0195773.

Shubhanshu Mishra and Vetle I. Torvik. 2016. Quantifying Conceptual Novelty in the Biomedical Literature. *D-Lib magazine : the magazine of the Digital Library Forum*, 22(9-10).

Michael J. Moravcsik and Poovanalingam Murugesan. 1975. Some results on the function and quality of citations. *Social Studies of Science*, 5(1):86–92.

Suchetha N. Kunnath, David Pride, Bikash Gyawali, and Petr Knoth. 2020. Overview of the 2020 wosp 3c citation context classification task. In *Proceedings of The 8th International Workshop on Mining Scientific Publications, ACM/IEEE Joint Conference on Digital Libraries (JCDL 2020*, Wuhan, China.

Shahzad Nazir, Muhammad Asif, Shahbaz Ahmad, Faisal Bukhari, Muhammad Tanvir Afzal, and Hanan Aljuaid. 2020. Important citation identification by exploiting content and section-wise in-text citation count. *PLOS ONE*, 15(3):1–19.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,

D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

D. Pride, P. Knoth, and J. Harag. 2019. Act: An annotation platform for citation typing at scale. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 329–330.

Xiaodan Zhu, Peter D. Turney, Daniel Lemire, and André Vellino. 2015. Measuring academic influence: Not all citations are equal. *CoRR*, abs/1501.06587.

Table 3: Per label model evaluation on the test data.

model	BACKGROUND	COMPARES CONTRASTS	EXTENSION	FUTURE	MOTIVATION	USES	accuracy	macro avg	weighted avg
Citing Context only (v1)									
lr	0.702	0.000	0.000	0.0	0.000	0.110	0.543	0.135	0.400
rf	0.692	0.042	0.032	0.0	0.018	0.058	0.528	0.140	0.396
gbt	0.683	0.057	0.056	0.0	0.000	0.110	0.518	0.151	0.400
mlp-3	0.641	0.202	0.022	0.0	0.031	0.219	0.467	0.186	0.412
mlp	0.639	0.206	0.049	0.0	0.028	0.198	0.462	0.187	0.410
All features (v2)									
lr	0.707	0.000	0.000	0.0	0.000	0.013	0.547	0.120	0.388
rf	0.698	0.075	0.000	0.0	0.000	0.045	0.535	0.136	0.397
gbt	0.700	0.071	0.000	0.0	0.000	0.114	0.534	0.148	0.408
mlp-3	0.663	0.175	0.000	0.0	0.059	0.165	0.492	0.177	0.414
mlp	0.649	0.176	0.065	0.0	0.060	0.163	0.478	0.185	0.411

Scubed at 3C task B - A simple baseline for citation context influence classification

Shubhanshu Mishra
shubhanshu.com
mishra@shubhanshu.com

Sudhanshu Mishra
Indian Institute of Technology Kanpur
Kanpur
India
sdhanshu@iitk.ac.in

Abstract

We present our team Scubed's approach in the 3C Citation Context Classification Task, Subtask B, citation context influence classification. Our approach relies on text based features transformed via tf-idf features followed by training a variety of simple models resulting in a strong baseline. Our best model on the leaderboard is a random forest classifier using only the citation context text. A replication of our analysis finds logistic regression and gradient boosted tree classifier to be the best performing model. Our submission code can be found at: https://github.com/napsternxg/Citation_Context_Classification.

1 Introduction

The number of research papers has increased exponentially in recent years. In order to efficiently access this scientific resource, we need automated solutions for extracting information from these records. Citations in research papers are important for multiple reasons e.g. comparing novelty (Mishra and Torvik, 2016), expertise (Mishra et al., 2018a), and self-citation patterns (Mishra et al., 2018b). For people new to the field, they are a way to increase knowledge whereas for experts in the field they act as useful pointers to summarize the paper. Citations are also used to measure various indexes which showcase the influence and reach of the researchers in their field. However, these indexes give equal weight to each citation. It has been established that all citations are not equal (N. Kunnath et al., 2020; Mishra et al., 2018b). In many cases, cited papers are used as examples or are not influential to the paper itself.

In this paper we describe our team, Scubed's entry for the citation context influence classification shared task (N. Kunnath et al., 2020). This work aims to develop models that can identify the influence of citations in the research papers, and hence can then be used to produce better indexes and make research more easily accessible to everyone.

1.1 Related Work

There has been a significant amount of work done in this area previously to better understand the significance of the citations in a paper (N. Kunnath et al., 2020). As the number of research papers increase with time, the algorithms for suggesting research papers become more and more important. These algorithms are a deciding factor for lots of measures of a researcher's influence in a field. The no. of citations of a paper are important for deciding measures such as h-index (Hirsch, 2005) and g-index (Egghe, 2006). These are influential measures for describing the significance of a researcher in a field. Scholars have argued that all of the citations in a paper should not have the same weight while determining the impact and reach of a paper. Moras et. al (Moravcsik and Murugesan, 1975) showed, that many references in research papers are redundant and quite often share little context with the citing paper. There have been many techniques for classifying citations as influential. However, one of the strongest baseline for this task is the prior citation count of the cited paper. Works of (Chubin and Moitra, 1975) show the effectiveness of citation count in determining influence. The work of (Zhu et al., 2015) points out suitable features for this task. They evaluated the performance of 5 classes of features, count, position, similarity, context and miscellaneous. They determined that counting the number of times a citation is referenced in a paper is the best estimator to determine the influence of a citation. (Hou et al., 2011) also showed that the count of a citation in a research paper is a simple and effective technique to assign its scientific contribution and influence.

(Nazir et al., 2020) applied SVM, Random Forests and Kernel Linear Regression classifiers to identify important and non-important citations. They used citation count and similarity scores using tf-idf features to train their models. Their results show that these techniques produce an improved precision score of 0.84 in these tasks.

2 Task and Data Description

This paper focuses on the WOSP 3C shared subtask B (N. Kunnath et al., 2020). In this sub-task, we were required to classify the citation context in research papers on the basis of their influence and purpose in the paper. For this shared task we used the ACL-ARC dataset(Jurgens et al., 2018) . The dataset consisted of 3000 labeled data-points annotated using the ACT platform (Pride et al., 2019). The data provided contains the following fields:

- Unique Identifier
- COREID of Citing Paper
- Citing Paper Title
- Citing Paper Author
- Cited Paper Title
- Cited Paper Author
- Citation Context
- Citation Class Label
- Citation Influence Label

To identify the citation being considered a #AUTHORTAG is placed in the citation. For this task the Citation Class Label field was ignored. This was a binary classification task, where the following target labels were used :

- **INCIDENTAL**
- **INFLUENTIAL**

To evaluate the models the macro-F1 score was used on the test data. The final score that was used to rank was not the public score but a different subset of data that was not visible to the participating teams. The teams were advised to make submissions that would perform the best overall and not just on the public subset.

3 Methodology

We utilize a simple approach based on text classification baseline methods. For the original submission we utilized a limited set of models. However, we trained additional models to conduct exhaustive evaluation for this paper. Below, we describe our workflow for pre-processing, feature extraction, and model-training.

3.1 Pre-Processing and Feature Extraction

The data provided was in raw text format which is not suitable for making predictions directly. In order to make useful predictions, it has to be first converted into numerical vector form that our models can process. The raw data consisted of columns having different attributes for which different feature extraction techniques had to be applied. For example, the *citing* and *cited title* consisted of a titles of the research papers whereas the *citation context* consisted of a description of the citation context. In order to efficiently process each column separately we used the *ColumnTransformer* module from the scikit-learn library (Pedregosa et al., 2011). Each of the column contained text data. To extract useful features from this text data we used the *TfidfVectorizer* from the scikit-learn (Pedregosa et al., 2011) library on each column. This generates the term frequency inverse document frequency(*tf-idf*) score for each of the texts in each column. The tf-idf score is a normalized count for the words occurring in the corpus. This type of feature however does not account for the position and inter-dependence of words. The tf-idf score is calculated as follows:

$$tf - idf(t, d) = tf(t, d) * idf(t) \tag{1}$$

$$idf(t) = \log\left(\frac{1+n}{1+df(t)}\right) + 1 \tag{2}$$

In the above equations, *tf* stands for term frequency which refers to the number of times a term *t* occurs in a document *d*. The *n* in **(2)** refers to the total number of documents present in the document set. *(Df(t))* refers to the document frequency which calculates the number of documents in the document set that contain the term *t*. The tf-idf score is a better feature compared to the count of words in a sentence. The tf-idf score down weights uninformative words like pronouns compared to more rare but informative words present in the document.

In the end we ended up using two version of text features for our models:

1. **Citing Context only (v1)**: uses only features extracted from *citation context* column. Our

hypothesis here is that citation context should have the highest signal for identifying how the citation is used.

2. **All features (v2)**: uses features extracted from *citation context* as well as *citing* and *cited title* column. Our hypothesis here is that using the combination of features from both citing and cited paper should improve the signal for identifying how the citation is used. However, we are also aware that this may also increase the proportion of noisy features.

3.2 Prediction Models

For this shared task we were allowed to submit a maximum of 5 models for evaluation on test data [1]. Our goal was to investigate usage of the most simple models based on proven linear and non-linear models which are faster and easier to train and deploy compared to the recent more powerful but resource hungry deep learning models. The following models were submitted for evaluation:

- **Logistic Regression Classifier (LR)**: A simple logistic regression model trained on the tf-idf features of 3 columns.
- **Random Forest (RF)**: Random Forest model with 100 trees in the forest and boot-strapping trained on the tf-idf features.
- **Gradient Boosting Classifier (GBT)**: A gradient boosted classifier with 100 boosting stages trained on the tf-idf features.
- **Multi-layer Perceptron Classifier (MLP)**: A 1 hidden layer multi-layer perceptron classifier with 100 nodes and Relu activation, optimized using Adam optimizer with a learning rate of 0.001 and momentum of 0.99.
- **Multi-layer Perceptron Classifier (MLP-3)**: A 3 hidden layer multi-layer perceptron classifier with 256, 256, and 128 nodes in the first, second and third layers with Relu activation optimized using Adam optimizer with a learning rate of 0.001 and momentum of 0.99.

All the models were trained using the scikit-learn library.

4 Results

Table 1 shows the the public and private leader board scores for each of our submissions for this task. Our RF (v1) model performed best on the leader-board while being quite similar to the top performing model (within 0.003 F1 score).

Table 1: Results for the Influence Sub-task. The overall best model used 116 submissions on the test data while we only utilized max 5 submissions as specified by the competition.

S.No	Model	Private	Public	Rank
1	LR (v2)	0.323	0.305	-
2	GBT (v2)	0.524	0.565	5
3	RF (v1)	**0.552**	0.591	**2**
4	MLP-3 (v2)	0.482	0.516	-
6	**Best**	**0.556**	0.576	**1**

4.1 Replication model performance after leader board submission

After the final leader board ranking, we decided to replicate the model performance on the actual test set provided to us by the shared task organizers. Our evaluation scores may not match with the submitted solutions as the model changes on each run and we did not record the random seed for the original submission. This analysis was conducted to generate comparable results for all models across the training and test sets (see table 2), and to further inspect the performance of the model on each label (see table 3 and 4).

First, table 2 shows the evaluation scores of all the models on the test set. One consistent pattern emerges, v1 models which use only the citation context text as its feature, consistently perform much better than v2 models. Next, the best v1 models are RF and LR. However, for v2, the best models is GBT which has consistent performance across v1 and v2. It appears that inclusion of extra features leads to over-fitting which is also evident from the training evaluation scores. Finally, the LR model (which is a linear model compared to all the other non-linear models) has the highest drop in evaluation score from v1 to v2, this may indicate that the linear model suffers more with the inclusion of noisy features.

Second, in table 3 we investigate the per label evaluation (in terms of F1 score) for each of the models. For both v1 and v2 features almost all models show similar performance on both labels. The only exception is the LR model which has 0.0 F1 score on Influential label for v2 features. Overall,

[1] https://www.kaggle.com/c/3c-shared-task-influence/rules

Table 2: Model evaluation scores (macro F1) on the test data on retraining models after leader board ranking.

model	v1		v2	
	test	train	test	train
mlp	0.523	0.992	0.494	1.000
mlp-3	0.524	0.994	0.496	1.000
gbt	0.535	0.770	*0.537*	0.804
rf	0.550	0.976	0.492	0.985
lr	**0.551**	0.830	0.314	0.343

it appears that these baseline models are quite good at learning this task compared to other submissions, while being fast and easy to implement.

Finally, in table 4 we list the top features for each class as identified based on the coefficients of the LR v2 model. Since, this is a binary classification task the model only learns a single coefficient for each feature. Hence, coefficients with negative values indicate features more important for the Incidental class while the coefficients with positive values indicate features more important for the Incidental class. The top features for influential label appear to be presence of words like **first**, while for incidental label it is **including**. The word **first** is a strong indicator of the citing paper being influential by being the first to introduce a concept. This phenomenon has also been observed in case of (Mishra and Torvik, 2016) which showed that novel papers (papers which were among the first to introduce a concept) are slightly more cited.

5 Discussion

Our results show that tradition tf-idf features give good performance for this shared task resulting in a strong baseline to compare against. Simple machine learning models like logistic regression, random forests, and gradient boosted trees perform well for this task compared to other submissions. Furthermore, the citation context contains the maximum signal for predicting citation influence. We were able to achieve one of the top performances in the task within the number of submissions required in the task. Due to the small dataset, multiple submissions increase the likelihood of the models to over-fit to the test set. Furthermore, our methods show that deep learning methods (e.g. mlp and mlp-3) do not give significant advantage over simpler machine learning methods. The minor loss in performance is acceptable compared to the increased speed and low computation of simple machine learning models.

Further analysis reveals that MLP based models are indeed over-fitting to the training data as shown by near perfect F1-score on the training data (see 2). Additionally, GBT models consistently achieve much better performance on the test set compared to other models, including RF model which was our best entry on the leader board. Furthermore, the highest performing label is the Influential label. All models (except LR) perform the worse on the Incidental when using all text features but when only using citation context, the label performance is similar across labels.

6 Conclusion

Our team 'Scubed' submitted 5 models for the citation context classification based on influence task. Out of the submitted models the random forest classifier performed the best on the test set achieving second position in this task. It achieved a private score of **0.55204** on the test set which was not only 0.003 behind the best performing model. We were able to achieve competitive results under minimum trials using fast and computationally cheap machine learning models.

References

Daryl E. Chubin and Soumyo D. Moitra. 1975. Content analysis of references: Adjunct or alternative to citation counting? *Social Studies of Science*, 5(4):423–441.

Leo Egghe. 2006. Theory and practise of the g-index. *Scientometrics*, 69(1):131–152.

Jorge Hirsch. 2005. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102:16569–72.

Wen-Ru Hou, Ming Li, and Deng-Ke Niu. 2011. Counting citations in texts rather than reference lists to improve the accuracy of assessing scientific contribution. *BioEssays*, 33(10):724–727.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association of Computational Linguistics*.

Shubhanshu Mishra, Brent D. Fegley, Jana Diesner, and Vetle I. Torvik. 2018a. Expertise as an aspect of author contributions. In *WORKSHOP ON INFORMETRIC AND SCIENTOMETRIC RESEARCH (SIG/MET)*, Vancouver.

Table 3: Per label model evaluation on the test data.

model	INCIDENTAL	INFLUENTIAL	accuracy	macro avg	weighted avg
Citing Context only (v1)					
mlp	0.487	0.559	0.526	0.523	0.526
mlp-3	0.512	0.535	0.524	0.524	0.525
gbt	0.568	0.502	0.537	0.535	0.532
rf	0.545	0.554	0.550	0.550	0.550
lr	0.567	0.536	0.552	0.551	0.550
All features (v2)					
lr	0.627	0.000	0.457	0.314	0.287
rf	0.489	0.495	0.492	0.492	0.492
mlp	0.469	0.519	0.495	0.494	0.496
mlp-3	0.444	0.548	0.501	0.496	0.500
gbt	0.499	0.575	0.540	0.537	0.540

Table 4: Top features in the LR (v1) model

	INCIDENTAL		INFLUENTIAL	
	feature	weight	feature	weight
0	including	-0.703	the	1.547
1	learning	-0.702	first	0.813
2	11	-0.652	were	0.742
3	2002	-0.629	to	0.676
4	and	-0.624	of	0.631
5	amp	-0.623	cessation	0.620
6	academic	-0.608	us	0.575
7	impact	-0.580	avh	0.518
8	13	-0.544	virus	0.513
9	research	-0.495	temperature	0.510

Shubhanshu Mishra, Brent D. Fegley, Jana Diesner, and Vetle I. Torvik. 2018b. Self-citation is the hallmark of productive authors, of any gender. *PLOS ONE*, 13(9):e0195773.

Shubhanshu Mishra and Vetle I. Torvik. 2016. Quantifying Conceptual Novelty in the Biomedical Literature. *D-Lib magazine : the magazine of the Digital Library Forum*, 22(9-10).

Michael J. Moravcsik and Poovanalingam Murugesan. 1975. Some results on the function and quality of citations. *Social Studies of Science*, 5(1):86–92.

Suchetha N. Kunnath, David Pride, Bikash Gyawali, and Petr Knoth. 2020. Overview of the 2020 wosp 3c citation context classification task. In *Proceedings of The 8th International Workshop on Mining Scientific Publications, ACM/IEEE Joint Conference on Digital Libraries (JCDL 2020*, Wuhan, China.

Shahzad Nazir, Muhammad Asif, Shahbaz Ahmad, Faisal Bukhari, Muhammad Tanvir Afzal, and Hanan Aljuaid. 2020. Important citation identification by exploiting content and section-wise in-text citation count. *PLOS ONE*, 15(3):1–19.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

D. Pride, P. Knoth, and J. Harag. 2019. Act: An annotation platform for citation typing at scale. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 329–330.

Xiaodan Zhu, Peter D. Turney, Daniel Lemire, and André Vellino. 2015. Measuring academic influence: Not all citations are equal. *CoRR*, abs/1501.06587.

Amrita_CEN_NLP @ WOSP 3C Citation Context Classification Task

Premjith B, Soman K.P
Center for Computational Engineering and Networking (CEN)
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
b_premjith@cb.amrita.edu, kp_soman@amrita.edu

Abstract

Identification of the purpose and influence of citation is significant in assessing the impact of a publication. '3C' Citation Context Classification Task in Workshop on Mining Scientific Publication is a shared task to address the aforementioned problems. This working note describes the submissions of Amrita_CEN_NLP team to the shared task. We used various supervised learning algorithms for the for classification of sentences encoded into a vector of dimension 300 generated using Word2vec model.

1 Introduction

The number of publications in the scientific domain increased exponentially recently, which allows researchers to look for various literature to extend their research. One method of finding the more relevant literature is to check the number of citations. A publication with more citation generally has more influence in the research community. Such publications typically give significant insight into specific problems. To test whether a paper is relevant for a particular domain, one should analyse the context in which it is written. It is equally important to identify the context of the citations also. This information, as well as the citation count, give a better understanding of a publication in a particular domain. In (Pride and Knoth, 2017), David Pride and Petr Knoth proposed an automatic method for identifying the citations with influence. In addition to it, identification of the purpose of a citation is also an essential task.

This paper describes the submission of Amrita_CEN_NLP team in '3C' Citation Context Classification Task a part of Workshop on Mining Scientific Publications (WOSP) 2020 (N. Kunnath et al., 2020). This shared task consisted of two subtasks. The goal of Subtask-A was to identify the purpose of the citations. The Subtask-B intended to classify the classification based on their importance into either influential or incidental. We used machine learning-based models for identifying the purpose and influence of the citations according to the context. The Word2Vec (Mikolov et al., 2013b), (Mikolov et al., 2013a) algorithm was used to convert the words into vectors by capturing the contexts of words in the given corpus. We employed various classification algorithms with varying dimensions of word vectors for the aforementioned tasks. The Random Forest classifier (Liaw et al., 2002), (Premjith et al., 2019a), (Premjith et al., 2019b) with a word vector of size 300 achieved the best performance with 5-fold cross-validation.

The organization of the paper as follows: Section 2 gives an brief description on the related research, which will be followed by a description of the dataset in the Section 3. The next section discusses the steps involved in designing the machine learning model and the paper concludes with the Conclusion section.

2 Related Works

The number of research works reported for the classification of scientific literature according to the context in which it is written are limited despite it's significance. S. Teufel et al. (Teufel et al., 2006) proposed an annotation scheme along with a classification model for the categorization of the citations. They considered 12 classes for annotation. The machine learning model was trained over 2829 citation instances collected from 116 articles. They used IBK algorithm for classification with hand-engineered features like cue phrases. D. Jurgens (Jurgens et al., 2018) used feature such as pattern-based features, topic-based features, and prototypical argument features to classify the documents into 6 classes. The authors used Random

Forest algorithm for classification. A. Cohan et al. (Cohan et al., 2019) used Glove and ELMO word embedding features and Bidirectional LSTM with attention models for classifying the citations.

3 Dataset description

The training and test datasets used for Subtask-A and Subtask-B were the same. The training data and test data contained 3000 and 1000 sentences, respectively. The Subtask-A was a multiclass problem with six classes in it. The distribution of the data for this task was highly uneven. In the dataset, 54.93% of the data belong to the BACKGROUND category, whereas the share of the FUTURE category was mere 2.07%. The Subtask-B was a binary classification problem, and the data set for this task contained evenly distributed class labels.

4 Dataset description

The training and test datasets used for Subtask-A and Subtask-B were the same. The training data and test data contained 3000 and 1000 sentences, respectively. The Subtask-A was a multi-class problem with six classes in it. The distribution of the data for this task was highly uneven. In the dataset, 54.93% of the data belong to the BACKGROUND category, whereas the share of the FUTURE category was mere 2.07%. The Subtask-B was a binary classification problem, and the data set for this task contained evenly distributed class labels.

5 System description

Amrita_CEN_NLP team participated in both the subtasks. We used machine learning algorithms for both tasks. The implementation pipeline is as follows,

1. Preprocessing
2. Feature representation
3. Classification and Result analysis

5.1 Preprocessing

The first step in preprocessing was to remove the unwanted characters. Therefore, we removed all the characters other than alphabets and digits from the text. It is followed by converting all the characters into lower case. From this text, all the stop words were removed.

5.2 Feature representation

This work utilized the Word2Vec algorithm for representing the words as vectors. Initially, the pre-trained model, namely "word2vec-google-news-300" was used for generating the word vectors. But the pre-trained model didn't yield any good results with classification algorithms. Therefore, we decided to construct the vector representation out of the training and testing data. The input data for Word2Vec was constructed by combining both training and test data. We experimented with different embedding dimensions with Continuous Bag-of-Words training approach. The context window size was set to 5. The minimum number of occurrence of each word to be considered for word vector generation was assigned to 1 to make sure that all the words in the corpus will find a representation. The embedding dimensions considered for the experiment were 50, 100 and 300.

The sentence vector was constructed by taking the linear combination of the word vectors, where the coefficients were assigned to one.

5.3 Method

We used machine learning algorithms such as Decision Tree, Random Forest, K-Nearest Neighbor (KNN), AdaBoost, and Logistic Regression for classification, and analyzed their performance. The ultimate goal of the classification algorithms is to find the optimal parameters, which depends on the proper tuning of the hyper-parameters. To find the optimal set of hyper-parameters for a classifier for each subtask, we utilized GridSearchCV() defined in the scikit-learn Python package (Pedregosa et al., 2011). This function finds the best combination of hyper-parameters by implementing 5-fold cross-validation. This process was repeated for each classifier with different word embedding dimension. Table 1 shows the hyper-parameters used for tuning all the classifiers and the optimal parameters obtained after the hyper-parameter tuning. The first value in the third column represents the optimal hyper-parameter values used for Subtask-A, and the second value is used for Subtask-B. The best estimator was used for training the data and fixed the performance by again cross-validating with 5-folds. The imbalance in the dataset used for Subtask-A may cause the classifier to predict the class labels for test data biased towards the BACKGROUND class because of its percentage of share in the dataset.

Classifier	Parameter	Parameter value	Optimal value
Decision Tree	Splitting criterion	gini, entropy	entropy, gini
	Splitter	best, random	random, random
Random Forest	# Estimators	50,100,150	100, 50
	Splitting criterion	gini, entropy	gini, gini
	Maximum features	auto, sqrt, log2	sqrt, log2
KNN	# Neigbours	3,5,7	7, 5
	Weights	uniform, distance	uniform, uniform
	Algorithm	auto, ball_tree, kd_tree, brute	auto, auto
Adaboost	Learning rate	0.01, 0.1, 1, 10, 100	0.01, 0.1
	Algorithm	SAMME, SAMME.R	SAMME, SAMME
Logistic Regression	Penalty	l1, l2, elasticnet, none	l1 , l1
	C	0.01, 0.1, 1, 10, 100	0.01 , 0.01
	Solver	newton-cg, lbfgs, liblinear, sag, saga	liblinear , saga
	Multi class	auto, ovr, multinomial	auto , auto

Table 1: Set of hyperparameters used for training the classifiers

Classifier	Embedding dimension		
	50	100	300
Decision Tree	36.76	33.49	35.63
Random Forest	47.73	48.56	54.93
KNN	48.76	48.13	50.00
Adaboost	54.93	54.93	54.93
Logistic Regression	54.93	54.93	54.93

Table 2: Cross-validated results for identifying the purpose of citations

Classifier	Embedding dimension		
	50	100	300
Decision Tree	49.87	50.03	49.63
Random Forest	48.07	48.77	54.83
KNN	50.23	49.27	52.26
Adaboost	52.26	52.27	50.33
Logistic Regression	52.37	52.40	53.03

Table 3: Cross-validated results for identifying citation influence

The performances of the cross-validated models were evaluated using the accuracy score. The evaluation scores of identifying the purpose and influence of citations are given in Table 2 and Table 3

5.4 Result Analysis

From the Tables 2 and 3, it is clear that the feature vector with dimension 300 achieved the highest accuracy in both the tasks. For the Subtask-A, AdaBoost, Random Forest and Logistic Regression obtained the maximum classification accuracy of 54.93%. For the Subtask-B, Random Forest attained the highest accuracy of 54.83%. Therefore, we decided to submit the Random Forest model for both the shared tasks.

Performance of the models with test data was evaluated using F1-score (macro). Tables 4 and 5 show the public and private macro F1-scores. For identifying the purpose of citation, the Decision Tree algorithm achieved the highest public F1-score of 0.2071, whereas Logistic Regression obtained the private highest F1-score of 0.1953. Random Forest reported the highest public as well as private F1-scores for identifying he citation influence task.

6 Conclusion

This working note reports the submission of the team Amrita_CEN_NLP for both Subtask-A and

Classifier	Public	Private
Decision Tree	0.2071	0.1673
Random Forest	0.1780	0.1398
KNN	0.1662	0.1356
Adaboost	0.1205	0.1149
Logistic Regression	0.1731	0.1953

Table 4: Public and private F1-score (macro) for identifying the purpose of citations with best classifier

Table 5: Public and private F1-score (macro) for identifying citation influence

Classifier	Public	Private
Decision Tree	0.4757	0.4760
Random Forest	0.4894	0.5153
KNN	0.4639	0.4377
Adaboost	0.3046	0.3224
Logistic Regression	0.3125	0.3258

Subtask-B. We experimented with different classifiers and different word embedding dimensions for identifying the best model for the classification. The cross-validated results showed that Random Forest classifier with 300 dimension Word2Vec features achieved the highest accuracy for both shared tasks.

References

Arman Cohan, Waleed Ammar, Madeleine Van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. *arXiv preprint arXiv:1904.01608*.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomforest. *R news*, 2(3):18–22.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Suchetha N. Kunnath, David Pride, Bikash Gyawali, and Petr Knoth. 2020. Overview of the 2020 wosp 3c citation context classificationtask. In *The 8th International Workshop on Mining Scientific Publications*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

B Premjith, KP Soman, M Anand Kumar, and D Jyothi Ratnam. 2019a. Embedding linguistic features in word embedding for preposition sense disambiguation in englishmalayalam machine translation context. In *Recent Advances in Computational Intelligence*, pages 341–370. Springer.

Bhavukam Premjith, Kutti Padannayl Soman, and Prabaharan Poornachandran. 2019b. Amrita_cen@ fact: Factuality identification in spanish text. In *IberLEF@ SEPLN*, pages 111–118.

David Pride and Petr Knoth. 2017. Incidental or influential?–a decade of using text-mining for citation function classification.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006. Automatic classification of citation function. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 103–110.

Overview of the 2020 WOSP 3C Citation Context Classification Task

Suchetha N. Kunnath
KMi, The Open University
Milton Keynes
UK
snk56@open.ac.uk

David Pride
KMi, The Open University
Milton Keynes
UK
david.pride@open.ac.uk

Bikash Gyawali
KMi, The Open University
Milton Keynes
UK
bikash.gyawali@open.ac.uk

Petr Knoth
KMi, The Open University
Milton Keynes
UK
petr.knoth@open.ac.uk

Abstract

The 3C Citation Context Classification task is the first shared task addressing citation context classification. The two subtasks, A and B, associated with this shared task, involves the classification of citations based on their purpose and influence, respectively. Both tasks use a portion of the new ACT dataset, developed by the researchers at The Open University, UK. The tasks were hosted on Kaggle, and the participated systems were evaluated using the macro f-score. Three teams participated in subtask A and four teams participated in subtask B. The best performing systems obtained an overall score of 0.2056 for subtask A and 0.5556 for subtask B, outperforming the simple majority class baseline models, which scored 0.11489 and 0.32249, respectively. In this paper we provide a report specifying the shared task, the dataset used, a short description of the participating systems and the final results obtained by the teams based on the evaluation criteria. The shared task has been organised as part of the 8th International Workshop on Mining Scientific Publications (WOSP 2020) workshop.

1 Introduction

Citation analysis for research evaluation has been a subject of interest for the past several decades. The conventional one dimensional perspective of citation analysis, based on the pure citation frequency, which treats all citations equally, has endured a lot of criticism way back [Moravcsik and Murugesan, 1975, Kaplan, 1965]. Subsequently, researchers have emphasised the need for developing new methods that consider the different aspects of the citing sentences. One such qualitative way for measuring the scientific impact is to analyse the citation context for discovering the author's reason for citing a paper. The text containing the reference to the cited document, the citation context, has proved to be a valuable signal for characterising the citation intent [Teufel et al., 2006]. The increase in the accessibility of the scientific publications, as well as the availability of full text of the research documents, from various services like CORE [Knoth and Zdrahal, 2012] facilitates the possibility of exploring citation contexts, thereby further extending the bibliometric studies for research assessment [Pride and Knoth, 2017].

Understanding the intent of citation has an essential role in measuring the scientific impact of the research papers. The possibility of knowing why a citation is included in one's work and how influential it is offers an excellent measure for evaluating the impact of a scientific publication. Previous approaches for citation context classification employed a variety of annotation schemes ranging from low to high granularity. Due to the lack of standard methods and annotation schemes, a comparison of the earlier systems is practically difficult. Earlier systems used datasets with very limited size and this is probably because of the difficulties in manually annotating the citation contexts. Besides, most of the research on citation context classification is not extensive enough and mainly reduced to specific domains of application, for instance, computer science and biomedical fields. This raises questions related to the generalisability of the presented models.

The 3C Shared task aims to create a platform encouraging researchers to participate in research in this area so that we can more reliably measure the performance of methods that have been tried in this area, establish the state-of-the-art and understand what works and what doesn't. Two subtasks associ-

ated with this shared task provide the participating teams the possibility to explore the new Academic Citation Typing (ACT) dataset [Pride et al., 2019, Pride and Knoth, 2020] for analysing the citation context and classify the associated citations based on their purpose (subtask A) and influence (subtask B). A total of four teams participated in subtask A, and five teams participated in the subtask B. We used Kaggle InClass competitions[1] for organising this shared task and the participating systems were evaluated using the macro f-score.

This overview paper presents the 2020 3C Shared Task organisation. Section 2 describes the related work; Section 3 discusses the shared task setup, the data used, the baselines, followed by task evaluation in Section 4. Section 5 summarises the participating system description. Section 6 and 7 presents the results and the conclusion.

2 Related Work

Several supervised machine learning based frameworks that inspect the language used in scientific discourse have been developed in the past to categorise citations based on their context. [Teufel et al., 2006] used an annotation scheme with 12 categories and applied machine learning techniques on 2,829 citation contexts from 116 articles, using linguistic features including the cue phrases. These 12 classes belonged to four top-level categories; citations explicitly mentioning weakness, citations that compares or contrasts, citations which agrees or uses or is compatible with the citing work and finally a neural class. A more fine-grained classification scheme introduced by Jurgens et.al [Jurgens et al., 2018] contains six categories and 1,941 instances from papers in Computational Linguistics(ACL-ARC dataset). The authors applied three novel features: pattern-based, topic-based and prototypical argument-based features besides the structural, lexical and grammatical, field and usage features.

The above mentioned approaches all used hand-engineered features for classification. [Cohan et al., 2019] proposed a neural multi-task learning method using non-contextualised (GloVe) and contextualised word embeddings (ELMo) along with BiLSTM and attention mechanism for citation intent classification. To achieve multi-task learning, the authors used two auxiliary tasks to aid the main classification task. The new dataset (SciCite) [Cohan et al., 2019] contains 11,020 instances belonging to Computer Science and Medicine domains and only three citation categories. A pre-trained model using 1.14M papers from Semantic Scholar[2], called SciBERT [Beltagy et al., 2019], was released in 2019 and achieved a macro f-score of nearly 85% with fine-tuning using the SciCite dataset.

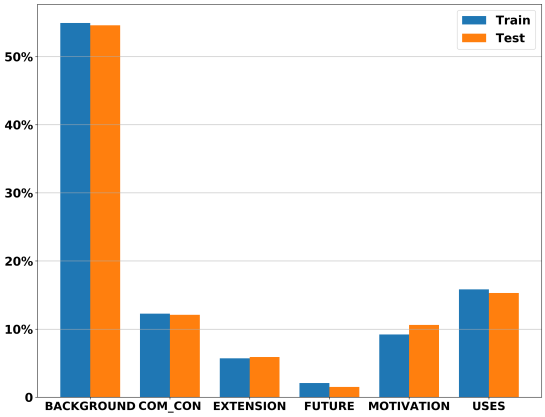


Figure 1: Subtask A data distribution

3 The 3C Shared Task

To address the limitations of citation context classification from the previous studies, we introduce a unified task to compare different citation classification methods on the same dataset. The shared task for the citation context classification, called, the "3C Shared Task", is organised as part of the International Workshop on Mining Scientific Publications (WOSP), 2020[3], collocated with the Joint Conference on Digital Libraries (JCDL) 2020[4]. As organisers, we believe, this shared task will provide the opportunity for comparing different classification systems and help progress the state-of-the-art. The competing systems in the 3C shared task will serve as a standard benchmark for future research in this direction.

3.1 Task Definition

The 3C shared task is a classification challenge, where each citation context is categorised based on its purpose and influence. The following are the output categories associated to the two subtasks respectively.

[1] https://www.kaggle.com/c/about/inclass

[2] https://www.semanticscholar.org/

[3] https://wosp.core.ac.uk/jcdl2020/index.html

[4] https://2020.jcdl.org/

unique_id	CC10
core_id	158977742
citing_title	Ontology-Based Recommendation of Editorial Products
citing_author	Thiviyan Thanapalasingam
cited_title	Ontological user profiling in recommender systems
cited_author	Middleton
citation_context	The main advantages of these solutions are i) the ability to exploit the domain knowledge for improving the user modelling process, ii) the ability to share and reuse system knowledge, and iii) the alleviation of the cold-start and data sparsity problems [16,#AUTHOR_TAG].
citation_class_label	BACKGROUND
citation_influence_label	INCIDENTAL

Table 1: ACT data format

- **Subtask A**: Multiclass classification of citation contexts based on purpose with categories - BACKGROUND, USES, COMPARES_CONTRASTS, MOTIVATION, EXTENSION, and FUTURE.
- **Subtask B**: Binary classification of citations into INCIDENTAL or INFLUENTIAL classes, i.e. a task for identifying the importance of a citation.

The shared task was managed and evaluated using the Kaggle InClass competitions, an easy to set up, free self-service platform for hosting Data Science challenges, with notebook support for GPU and code sharing. The ability to maintain a leaderboard, which allows the participants to view results immediately after submission, built-in evaluation metrics and automated submission scoring are some of the features offered by Kaggle.

Both subtasks were organised as separate competitions in Kaggle. The shared task homepage for subtask A can be found at https://www.kaggle.com/c/3c-shared-task-purpose/. The following url correspond to the competition page for the subtask B, https://www.kaggle.com/c/3c-shared-task-influence/. The task participants were required to:

- Develop methods to classify the citations based on its purpose or influence and submit the results via Kaggle
- Document and submit their method for classifying the citations as a short paper
- Provide source code for each method

The competitions lasted 43 days, starting from May 11, 2020 till June 22, 2020.

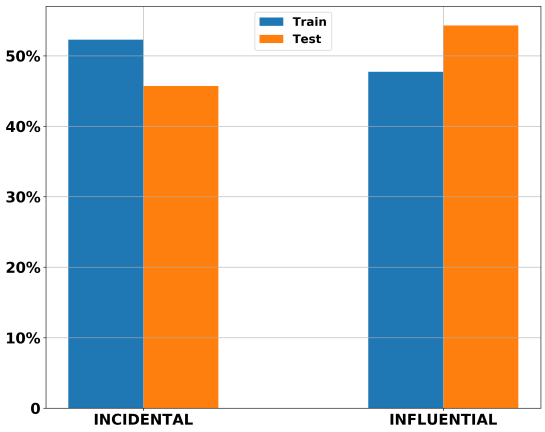


Figure 2: Subtask B data distribution

3.2 Dataset

The previous studies on citation classification systems used datasets that were annotated by domain experts and independent annotators, making the evaluation process relatively slow and expensive. Existing datasets in the field are, as a result, also confined to a specific domain, mainly computer science and biomedical domains, because this is the domain in which the annotators can could label the instances. The citation contexts need not always contain explicit signals that express the author's motivation for citing a paper. Since interpreting the citation intent is difficult for an independent annotator, authors themselves are in a better position to report their motivations in citing a paper [Pride and Knoth, 2020]. [Pride et al., 2019] used this strategy; asking authors to annotate their papers

for tagging citations based on their purpose and influence. The new dataset, called the ACT dataset is the largest multi-disciplinary dataset of its type in existence with annotations for 11,233 citations annotated by 883 authors [Pride and Knoth, 2020].

Table 1 illustrates a sample instance from the ACT dataset. Each citation context in the dataset contains the label, "#AUTHOR_TAG", which represents the citation that is being considered. The citing_title and citing_author corresponds to the details of the document with the citation context. The dataset also has information about the cited paper (tile and author details) corresponding to the #AUTHOR_TAG. The citation_class_label represents the purpose category and the citation_influence_label corresponds to the binary class based on how influential the citation is.

The participants were provided with a labeled training dataset in the csv format with 3,000 instances, annotated using the ACT platform. Since Kaggle InClass competitions doesn't allow hosting more than one task using the same interface, separate competitions had to be created. Also, we had to split the dataset into two, based on the citation class label and the citation influence label. We also converted the categorical labels to numeric values. The citation class labels corresponds to values between 0 and 5, where each value represents the following categories:

0 - BACKGROUND

1 - COMPARES_CONTRASTS

2 - EXTENSION

3 - FUTURE

4 - MOTIVATION

5 - USES

Similarly, the citation influence labels were represented with values 0 or 1, as follows:

0 - INCIDENTAL

1 - INFLUENTIAL

Figure 1 illustrates the data distribution for Subtask A. The dataset is highly imbalanced with nearly 55% of the instances belonging to the BACKGROUND class in the training set. The FUTURE class has the lowest number of instances with just 62 and 15 instances in the training and the test dataset, respectively. The number of instances of INCIDENTAL and INFLUENTIAL classes used for Subtask B is shown in Figure 2. The dataset is relatively less skewed for Subtask B, with the number of instances associated with the incidental class (1,568) being higher than the influential class (1,432) for the training set. For both tasks, we ensured that the data distribution of categories in training set to be nearly the same as the test set. Besides the ACT dataset, participants were also encouraged to use external datasets, like the ACL-ARC [Jurgens et al., 2018], which is compatible with our dataset, for training, provided, the teams mention this while describing the systems.

3.3 The Baseline

We made an initial submission based on a simple majority class prediction as a baseline entry for both subtasks. For Subtasks A and B, the majority class corresponds to the categories, BACKGROUND and INCIDENTAL, respectively. As the competition proceeded, we also made a submission based on the BERT model [Devlin et al., 2018]. We used the pre-trained model, scibert-scivocab-uncased[5], pretrained on a sample of 1.14M multi-domain papers from the Semantic Scholar [Beltagy et al., 2019]. The 3,000 training instances were then used for fine-tuning, to obtain the task-specific results. The rational here has been to test how a state-of-the-art method, recently reported in [Cohan et al., 2019] performs compared to the methods submitted by the participants.

4 Evaluation

The evaluation was based on the test set of 1,000 examples. The test dataset was partitioned into public and private sets in Kaggle. 50% of the test set was used for the initial evaluation, and the evaluation results against it appeared on the public leaderboard as the competition progressed. The rest of the data, which is the private partition on the test file, was used for the final scoring. The private leaderboard was visible only to the shared task organisers during the competition period.

We used macro f-score for evaluating the submissions.

$$F1 - macro = \frac{1}{n}\sum_{i=1}^{n}\frac{2 \times_{Pi} \times_{Ri}}{_{Pi}+_{Ri}} \quad (1)$$

where $_{Pi}$ and $_{Ri}$ denotes the precision and recall for class i and n represents the number of classes. We chose macro f-score in light of the disproportionate distribution of output categories in our dataset and to encourage the task participants to focus on the

[5] https://github.com/allenai/scibert

Team Name	Run ID	Leaderboard	
		Public	Private
UFMG	5	**0.21460**	**0.20560**
scibert		0.17966	0.19026
Scubed	3	0.17599	0.18146
Amrita_CEN_NLP	2	0.11981	0.12542
majority_class_baseline		0.12047	0.11489

Table 2: Public and private leaderboard macro f1-scores for citation context classification based on purpose (Subtask A)

Team Name	Run ID	Leaderboard	
		Public	Private
Paul Larmuseau	1	0.57556	**0.55565**
Scubed	3	**0.59108**	0.55204
UFMG	1	**0.59108**	0.54747
Amrita_CEN_NLP	2	0.48937	0.51534
scibert		0.54747	0.50012
majority_class_baseline		0.30458	0.32249

Table 3: Public and private leaderboard macro f1-scores for citation context classification based on influence (Subtask B)

detection of the minority classes, which are particularly crucial for advancing the field of research metrics beyond just counting citations.

The submission file, in csv format, contains the unique id followed by the citation class label for Subtask A or citation influence label for Subtask B. We encouraged team submissions in Kaggle and did not set any restrictions on the team size. The limit on the number of submissions per day was set to 20. All teams were allowed to submit a maximum of 5 runs to the competition for the final evaluation for each of the tasks. The best submitted system will be used by kaggle for final scoring on the private leaderboard.

5 Participating System Description

This section presents the overview of the systems used by the participated teams, UFMG, Paul Larmuseau, Scubed and Amrita_CEN_NLP in the 3C shared task. Except for Paul Larmuseau, rest of the teams participated in both the tasks. The teams that participated in both tasks used the same approach while making submissions to Subtask A and Subtask B.

5.1 UFMG

Team UFMG[6] explores the possibility of enhancing the results by using a combined text representations for capturing the statistical, topical and the contextual information. For this, they chose Term Frequency-Inverse Document Frequency (TF.IDF) for word representation (upto bigrams), Latent Dirichlet Allocation (LDA) for topic extraction from citation context and finally GloVe embeddings[7] to obtain the word vector representation for capturing the word co-occurrences. The team obtained the highest score of 0.2056 for subtask A by combining the above mentioned word representations for the passive aggressive classifier, an incremental learning mechanism. However, for Subtask B, UFMG obtained the best overall score of 0.54747, finishing as third on the leaderboard, just by using a single feature, TF.IDF. Furthermore, by using additional feature like self citation along with the TF.IDF, the team claims to have obtained a 3.1 % improvement in the final score for Subtask B [Valiense de Andrade and Goncalvesh, 2020].

5.2 Scubed

The team Scubed[8] applied TF.IDF on the columns, citing title, cited title and the citation context in the dataset. They used off-the-shelf machine learning based models, including Logistic Regression (LR), Random Forest (RF), Gradient Boosting Classifier (GBT) and two variants of the Multi-Layer Perceptron (MLP) classifiers. For Subtask A, the best performing model using MLP obtained a private score of 0.18146 and the team finished third. However, for the binary classification task, RF achieved the best score and the team finished second on the leaderboard with a macro f-score of 0.55204. The team also reports a per category model evaluation using the truth labels of the test set [Mishra and Mishra, 2020a,b].

5.3 Paul Larmuseau

The best system in the subtask B was that of Paul Larmuseau[9]. The team used a combined TF.IDF weighting and fasttext embedding, consisting of 1 million word vectors trained on Wikipedia 2017[10]. Another important feature used by the team was

[6] 10.6084/m9.figshare.12638807

[7] https://nlp.stanford.edu/projects/glove/

[8] 10.6084/m9.figshare.12638846

[9] 10.6084/m9.figshare.12638840

[10] https://fasttext.cc/docs/en/english-vectors.html

the cosine similarity, calculated between the citing title and a combination of cited title and the citation context. As part of the pre-processing step, they also experimented with feature scaling (based on the maximum absolute values) and dimensionality reduction (single value decomposition regression) techniques. The team experimented with different approaches and obtained the highest private score of 0.55566 using LR, finishing first in Subtask B [Larmuseau, 2020].

5.4 Amrita_CEN_NLP

The team Amrita_CEN_NLP[11] used Word2Vec for extracting the contextual information and feature representation. In order to built the vocabulary, the team used the shared task training and the test dataset. The team experimented with different classifiers like LR, Decision Tree (DT), k-Nearest Neighbour (k-NN), LR and Ada Boost. A cost sensitive learning approach for assigning separate weights was used for Subtask A, to address the class imbalance issue. The best score for both subtasks was achieved using RF [B and K.P, 2020].

6 Results

Table 2 shows the public and the private macro f-scores obtained by the teams for Subtask A. The highest public and private macro f-score was obtained by the team, UFMG. The submission based on scibert model scored the second best result with a private score of 0.19026. This was followed by the teams scubed and Amrita_CEN NLP in the third and fourth positions. All the teams substantially outperformed the majority class baseline classifier. Since the dataset for purpose classification task was highly skewed, with the majority of the classes belonging to the BACKGROUND class and the fact that we used macro f-score for evaluating the systems, all the systems submitted for this task scored less when compared to the Subtask B.

The results for the final evaluation of systems submitted for Subtask B is shown in Table 3. The highest performing system, submitted by Paul Larmuseau achieved a private macro f score of 0.55565, ranking as first for Subtask B. However, two other systems submitted by the teams Scubed and UFMG obtained an even higher score of 0.59108 on the public data. The deep learning based language model scibert achieved lesser score compared to the rest of the submissions using simpler machine learning model for this binary classification task. Not surprisingly, the systems submitted to Subtask B achieved better results when compared to the other task, because of the lesser number of categories and less skewness in the data distribution.

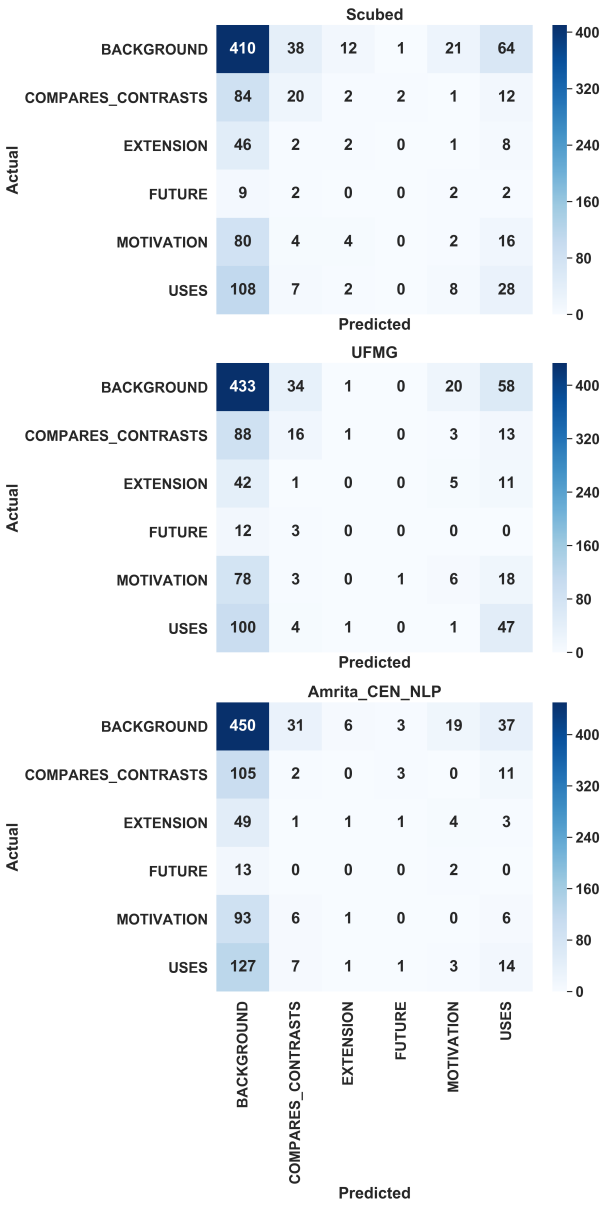


Figure 3: Confusion Matrix for subtask A

7 Discussion

The 3C Shared task is the first open competition for citation context classification. This shared task could be considered as a new benchmark for these tasks as we release both the data and the source code of all the submitted systems. All the teams that participated in this shared task used simple

[11] 10.6084/m9.figshare.12638849

Run ID	Team	Field Used	Model	Features	Public Score	Private Score
1	UFMG	citation_context	Passive Aggressive	TF.IDF	0.19829	0.19425
2				LDA	0.12923	0.15826
3				GloVe	0.12047	0.11489
4				TF.IDF+LDA	0.19124	0.19572
5				TF.IDF+GloVe	0.19945	0.20037
6				TF.IDF+LDA+GloVe	**0.21460**	**0.20560**
1	Scubed	citing_title, cited_title, citation_context	GBT	TF.IDF	0.15001	0.14381
2			RF	TF.IDF	0.14262	0.15826
3			MLPC	TF.IDF	**0.17599**	**0.18146**
1	Amrita_CEN_NLP	citation_context	DT	Word2Vec	**0.20709***	0.16732*
2			RF	Word2Vec	0.11981	0.12542
3			kNN	Word2Vec	0.16623*	0.13563*
4			Adaboost	Word2Vec	0.12047*	0.11489*
5			LR	Word2Vec	0.17309*	**0.19530***

* Post-Evaluation Results

Table 4: Overall Result (Subtask A)

Run ID	Team	Field Used	Model	Features	Public Score	Private Score
1	Paul Larmuseau	cited_title, citing_title, citation_context	LR	TF.IDF	**0.57556**	0.55565
2			LR	fasttext + TF.IDF	0.54726 *	**0.60333** *
1	UFMG	citation_context	Passive Aggressive	TF.IDF	**0.59108**	0.54747
2				LDA	0.30458	0.32249
3				GloVe	0.30458	0.32249
4				TF.IDF+LDA	0.32707	0.36156
5				TF.IDF+GloVe	0.30458	0.32249
6				TF.IDF+LDA+GloVe	0.30458	0.32249
7				TF.IDF+self_citation	0.57556 *	**0.55565** *
1	Scubed	citing_title, cited_title, citation_context	LR	TF.IDF	0.30458	0.32249
2			GBT	TF.IDF	0.56473	0.52351
3			RF	TF.IDF	**0.59108**	**0.55204**
4			MLP-3	TF.IDF	0.51589	0.48187
1	Amrita_CEN_NLP	citation_context	DT	Word2Vec	0.47565	0.47596
2			RF	Word2Vec	**0.48937**	**0.51534**
3			kNN	Word2Vec	0.46386	0.43769
4			Adaboost	Word2Vec	0.30458	0.32249
5			LR	Word2Vec	0.31250	0.32579

* Post-Evaluation Results

Table 5: Overall Result (Subtask B)

machine learning-based classifiers, including logistic regression, random forest, and multi-layer perceptron. One of the teams experimented with the online learning technique for faster computation. As with feature representation, the conventional approach used by the majority of the teams was TF.IDF. The prospect of employing word vectors developed using Wikipedia, the shared task dataset and the use of pre-trained embeddings like GloVe were explored by the teams.

Figure 3 shows the confusion matrix for the best systems submitted by the teams Scubed, UFMG, and Amrita_CEN_NLP for the subtask A. The most successfully classified category is BACKGROUND. The winning team, UFMG, classified nearly 80% of the BACKGROUND class instances correctly. The number of true positives for the minority class FUTURE is zero, which implies that none of the above mentioned teams could successfully categorise the instances to this class. The imbalanced nature of the subtask A dataset significantly affects the performance of the systems submitted by teams, which is one of the challenging aspects as far as citation function classification task is concerned.

Tables 4 and 5 displays the public and private scores obtained by teams for the different systems they submitted for subtask A and subtask B respectively. All the teams for both tasks used the data field, citation_context as the main source of semantic information for feature extraction, and classification. Two teams also examined citing_title and the cited_title fields for extracting useful features. Since Kaggle allows late submissions for the hosted competitions, the participants can still submit results to get better scores, although this will not be visible on the public and the private leaderboard. Both the tables also contain the post-evaluation results obtained by some of the teams.

The current deep learning based state-of-the-art language models like scibert could not achieve better results on our dataset, and as the leaderboard indicates, such sophisticated models are beaten by more simpler methods, that are significantly less computationally expensive on this task. One possible reason for this could be the lesser number of training instances we provided to the participants.

8 Conclusion

Citations, which act as a connection between the cited and the citing articles, cannot be treated equally and serve different purposes. Traditional citation analysis based on mere citation counts take into consideration just the quantitative factors. Analysing the citation context for classifying citations based on their function and influence has many applications and the most important being its implementation in the research quality evaluation. One of the greatest challenges faced in the citation context analysis for identifying the citation function and its influence is the absence of multi-disciplinary datasets and unavailability of medium to fine grained schemes which sufficiently captures information for citation classification [Hernández-Alvarez and Gómez, 2015]. Although previous works on the problem of citation context classification exist, lack of shared datasets, common conventions and annotation schemes caused the benchmarking of systems on the same tasks difficult.

The 3C Shared task constitutes the first systematic effort to a) compare different methods on the same data, b) on the same classification taxonomy across two previously reported tasks, and c) on multi-disciplinary data. We propose the unifying framework of the 3C shared task to be used as a standardised benchmark for this task, as we make all the submitted systems to this shared task, publicly available. We believe this will allow future comparison of participating systems head-to-head on the same data and task. The results obtained by the teams indicate the relevance of the simple machine learning based models over complex deep learning based approaches. The winning team for the subtask A, UFMG obtained an overall score of 0.19425. The team, Paul Larmuseau finished at first position on the leaderboard with a macro f score of 0.55565 for subtask B.

References

Premjith B and Soman K.P. Amrita_cen_nlp wosp 3c citation context classification task. In *The 8th International Workshop on Mining Scientific Publications, ACM/IEEE Joint Conference on Digital Libraries (JCDL 2020)*, Virtual - China, 2020.

Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

Arman Cohan, Waleed Ammar, Madeleine Van Zuylen, and Field Cady. Structural scaffolds for citation intent classification in scientific publications. *arXiv preprint arXiv:1904.01608*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and

Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Myriam Hernández-Alvarez and José M Gómez. Citation impact categorization: for scientific literature. In *2015 IEEE 18th International Conference on Computational Science and Engineering*, pages 307–313. IEEE, 2015.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406, 2018.

Norman Kaplan. The norms of citation behavior: Prolegomena to the footnote. *American documentation*, 16(3):179–184, 1965.

Petr Knoth and Zdenek Zdrahal. Core: three access levels to underpin open access. *D-Lib Magazine*, 18 (11/12):1–13, 2012.

Paul Larmuseau. Find influential articles in a dataset. In *The 8th International Workshop on Mining Scientific Publications, ACM/IEEE Joint Conference on Digital Libraries (JCDL 2020)*, Virtual - China, 2020.

Shubhanshu Mishra and Sudhanshu Mishra. Scubed at 3c task a - a simple baseline for citation context purpose classification. In *The 8th International Workshop on Mining Scientific Publications, ACM/IEEE Joint Conference on Digital Libraries (JCDL 2020)*, Virtual - China, 2020a.

Shubhanshu Mishra and Sudhanshu Mishra. Scubed at 3c task b - a simple baseline for citation context influence classification. In *The 8th International Workshop on Mining Scientific Publications, ACM/IEEE Joint Conference on Digital Libraries (JCDL 2020)*, Virtual - China, 2020b.

Michael J Moravcsik and Poovanalingam Murugesan. Some results on the function and quality of citations. *Social studies of science*, 5(1):86–92, 1975.

David Pride and Petr Knoth. Incidental or influential?– a decade of using text-mining for citation function classification. 2017.

David Pride and Petr Knoth. An authoritative approach to citation classification. In *2020 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, Virtual - China, 2020.

David Pride, Petr Knoth, and Jozef Harag. Act: an annotation platform for citation typing at scale. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 329–330, Urbana-Champaign, Illinois, 2019. IEEE.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. Automatic classification of citation function. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 103–110, 2006.

Claudio Moises Valiense de Andrade and Marcos Andrer Goncalvesh. Combining representations for effective citation classification. In *The 8th International Workshop on Mining Scientific Publications, ACM/IEEE Joint Conference on Digital Libraries (JCDL 2020)*, Virtual - China, 2020.