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.

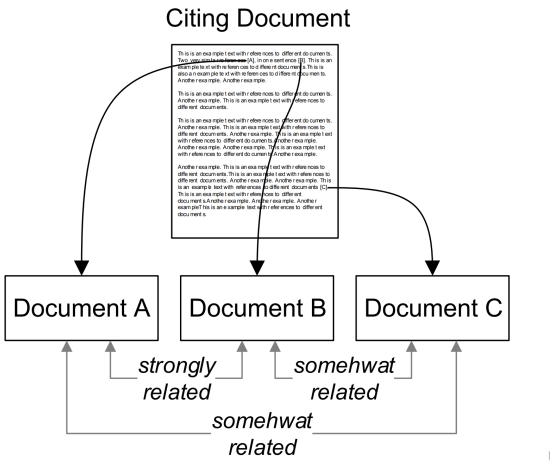

Figure 1: Illustration of Citation Proximity Analysis (Gipp and Beel, 2009). A citing document cites the three documents A, B, and C. Documents A and B are cited within the same sentence and are hence strongly related. Documents A and C, as well as documents B and C, are each cited within different paragraphs. 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

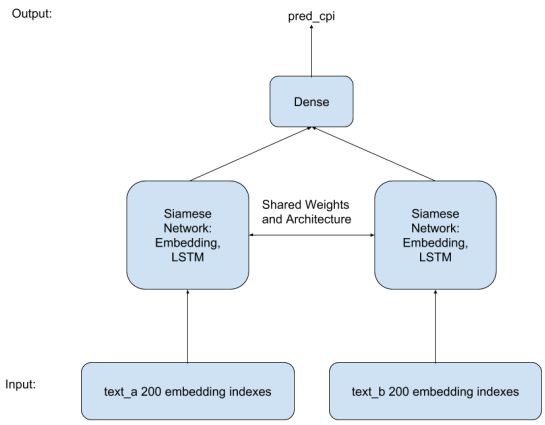

Figure 3: Siamese Neural Network Architecture Diagram.

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

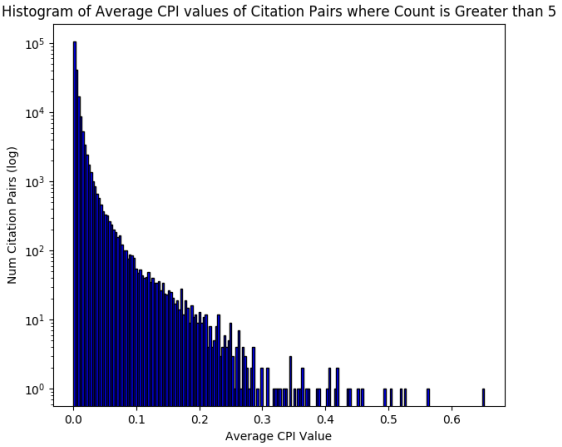

Figure 4: Distribution of CPI Values in the Wikipedia dataset. Many CPI values are very small.

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

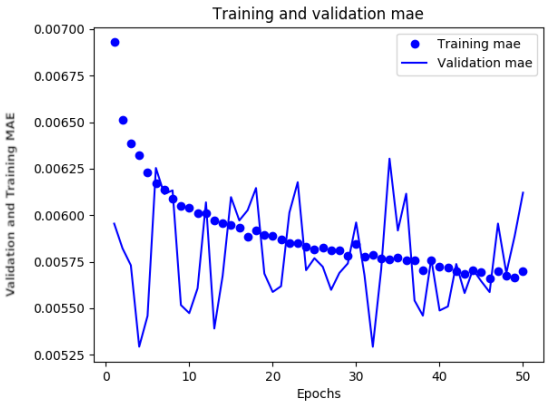

Figure 6: Mean Average Error of Siamese 1D CNN and LSTM over 50 Epochs.

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*.