CiteNet: A Search and Visualization Tool for Scientific Literature

Duncan T. Forster[1,3], **John M. Giorgi**[2,3]
University of Toronto, Toronto, ON, Canada
[1]Department of Molecular Genetics
[2]Department of Computer Science
[3]Donnelly Centre for Cellular and Biomolecular Research
duncan.forster@mail.utoronto.ca,
john.giorgi@utoronto.ca

Abstract

We present CiteNet, a search and visualization web application for exploring scientific literature. CiteNet provides an alternative to the current keyword-based search paradigm, offering instead a "key-paper" approach where the user implicitly specifies the search content by providing a set of related articles. By using underlying citation and semantic relationships between articles, CiteNet extracts a highly relevant set of articles related to the user's query. The semantics of each article is captured by a novel document embedding method. CiteNet implements a suite of visualization features to allow the user to explore the results, quickly identify important articles and further refine their search query. CiteNet is available at https://citenet.io.

1 Introduction

Searching for published literature is a crucial task for scientific researchers. Widely used search engines such as Google Scholar[1] and PubMed[2] are keyword-based, relying on the user to provide a query that is specific and unique enough to a) capture the relevant topic of interest and b) exclude the substantial space of false-positive results. This paradigm places a high burden on the user, requiring knowledge of esoteric keywords to yield the best search results. In addition to this, these search engines return results as ranked lists spread over many pages, resulting in a fragmented search experience since the user must keep track of relevant papers while navigating between pages.

Here we introduce CiteNet, a search and visualization web application designed to modernize and expedite literature search. CiteNet is familiar and intuitive, requiring no preamble or specialized knowledge to use. Underlying the search algorithm is a citation graph and a semantic graph, where articles (nodes) are connected through citation relationships (cites or cited by) and the semantic similarity of article abstracts, respectively. CiteNet relies on a novel "key-paper" querying approach, where the user specifies a set of relevant research articles (seeds) instead of keywords. Here, the shared concepts of the seed papers implicitly define the search. CiteNet attempts to return other, highly relevant research articles based on these shared concepts. Given the user's query, CiteNet traverses the citation and semantic graphs to identify these articles and displays them in an interactive visualization. Currently, CiteNet indexes approximately 23 million biomedical articles from the PubMed database and will be expanded in the future to index articles from all scientific disciplines.

In the following section (§2), we provide a brief walk-through on how to perform a search with CiteNet. In (§3), we highlight two compelling case studies. In (§4), we discuss CiteNet's implementation, namely its document embedding method, search engine and server architecture. We conclude with a summary and discussion of future directions (§5).

2 Using CiteNet

The CiteNet homepage presents users with a standard search engine interface (Figure 1A). The user can input article title, author name(s), journal or publication date and select the correct article from a dropdown. This process can be repeated to define a set of seed articles. After searching, users can view the returned results as a ranked list (Figure 1B) or a citation graph (Figure 1C). The rank view displays a simplified article view containing title and author(s) to the left and a dialogue containing more detailed information (such as abstract, journal title and publication date) to the right. The

[1]https://scholar.google.com/
[2]https://pubmed.ncbi.nlm.nih.gov/

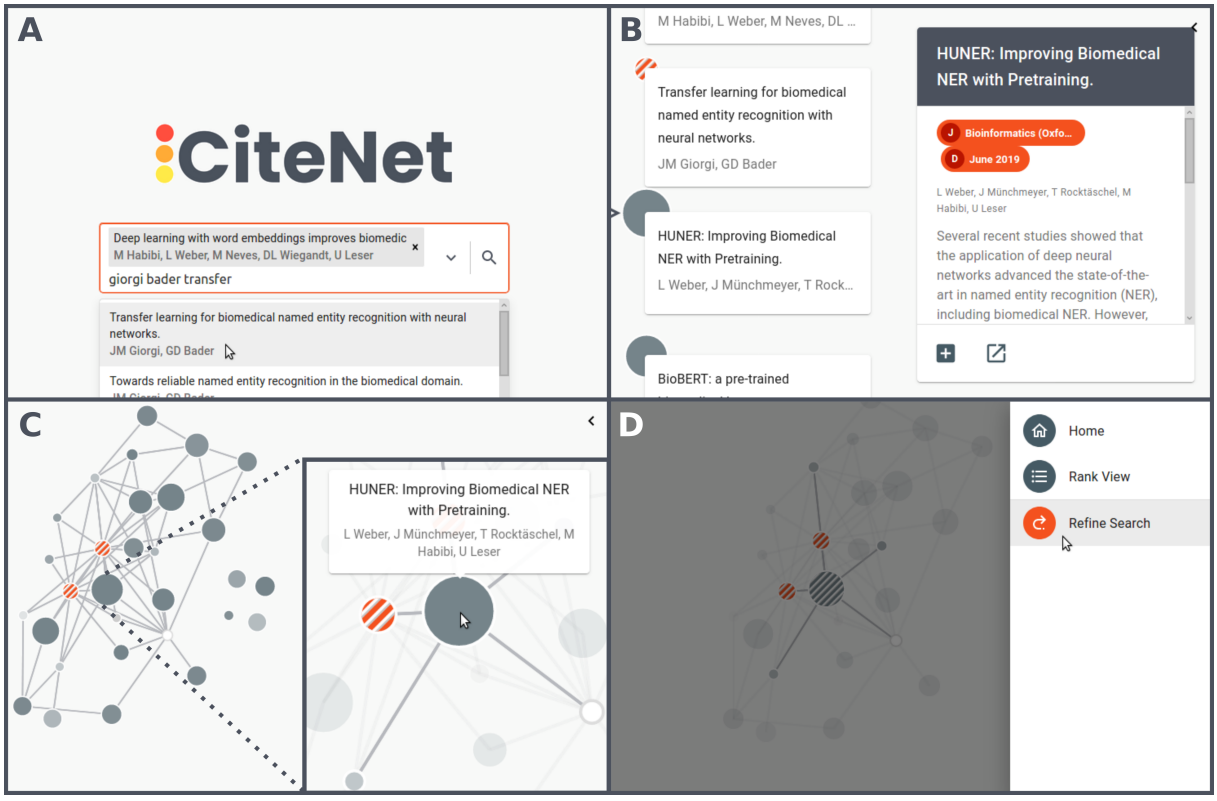

Figure 1: CiteNet user interface. **(A)** CiteNet homepage where users can input query papers using a dropdown listing papers from the database. **(B)** Rank view displaying the top search results. **(C)** Network view displaying citation relationships between nodes and **(inset)** popover and neighbor focusing when a node is hovered over. **(D)** Side panel with option to switch views or refine the search with the new search queue (denoted by hatched nodes). In **(B)**, **(C)** and **(D)** node size corresponds to relevance to the query articles. Lighter coloured nodes are older articles, orange nodes are the original query articles and nodes with a hatch pattern are articles in the current search queue.

network view allows users to intuitively identify communities of articles and visually interrogate citation relationships between them. When the user hovers over a node, a popover displays its title and authors and its direct citation neighbours are focused (Figure 1C, inset). Clicking on a node in network view brings up a modal identical to the rank view dialogue, containing detailed information about the given article. The user has the option to add additional articles to the search queue by clicking the "plus" icon in the dialogue (Figure 1B) and triggering a new search from a side panel (Figure 1D).

CiteNet's search paradigm relies on having multiple seed articles to produce the best results. Each article provided to the search algorithm implicitly contains a set of concepts. As more articles are added, the number of shared concepts necessarily decreases, refining the query. CiteNet aims to find papers that lie at this intersection of concepts. As the user navigates the search results, they have the option to add or remove any articles to their search query and perform the search again – allowing for iterative refining of the original query.

3 Case Studies

To qualitatively validate CiteNet's key-paper search paradigm, we conduct two case studies that focus on its most compelling features: semantically informed literature search and iterative search.

3.1 Semantically Informed Search

While the citation network serves as a human-curated set of relationships between articles, it is not always a reliable indicator of relevance. For example, methods (e.g. a machine learning algorithm), resources (e.g. a benchmark dataset) or review papers come to dominate the surrounding subgraph in the citation network. A search performed over this subgraph will return these highly cited papers even when they are not amongst the most relevant to a user based on their provided

Table 1: Titles of the top 5 most similar articles (descending order) for a given seed(s) when only the citation network is used to inform the search and when both the citation and semantic networks are used. Bold, seed paper(s).

	PMID	Text
Seed	28881963	**Deep Learning With Word Embeddings Improves Biomedical Named Entity Recognition**
Citation	23584833	Annotating the Biomedical Literature for the Human Variome
	24834132	Chemical Named Entities Recognition: A Review on Approaches and Applications
	25268232	Annotated Chemical Patent Corpus: A Gold Standard for Text Mining
	30600484	MADEx: A System for Detecting Medications, Adverse Drug Events, and Their Relations From Clinical Notes
	23413997	Gimli: Open Source and High-Performance Biomedical Name Recognition
Citation & Semantic	29718118	D3NER: Biomedical Named Entity Recognition,Using CRF-biLSTM Improved With Fine-Tuned Embeddings of Various Linguistic Information
	31243432	HUNER: Improving Biomedical NER With Pretraining
	27283952	TaggerOne: Joint Named Entity Recognition and Normalization With semi-Markov Models
	31138109	CollaboNet: Collaboration of Deep Neural Networks for Biomedical Named Entity Recognition
	30307536	Cross-type Biomedical Named Entity Recognition With Deep Multi-Task Learning

seed paper(s). A natural solution to this problem is to incorporate *semantic* information in the search, which may have the effect of down-weighting these highly cited papers and up-weighting less cited, but more relevant papers.

To this end, we perform a case study on the effect of incorporating semantic information into CiteNet's search algorithm. In short, we use a novel document representation technique to embed each article abstract into a continuous vector, where we expect semantically similar abstracts to end up close together in the embedding space. We then use these embeddings to form the semantic graph (see §4.1). Specifically, we chose a seminal paper on the application of deep neural networks to biomedical named entity recognition (NER) as our seed paper (Habibi et al., 2017), and explore the search results returned by CiteNet when a) only the citation network informs the search and b) both the citation and semantic networks inform the search.

Table 1 presents the results of these two searches. When only the citation network informs the search, four of the top five results present a particular tool or benchmark corpus for biomedical NER, and the remaining paper is a review on NER for chemical entities only. These results are in line with our intuition that highly cited methods, resources and reviews will come to dominate the results of a search based on the citation network only. When the semantic network additionally informs the search, CiteNet returns five highly relevant articles. Like the seed paper, each of the returned articles presents a methodological advance in applying deep learning to biomedical NER. Furthermore, four of the returned articles use deep learning models based on the same underlying architecture as the model used in the seed paper (Lample et al., 2016).

3.2 Iterative Search

Searching for scientific literature is an iterative process. A user may not know what they are looking for precisely at the outset but instead perform successive searches until arriving at a set of highly relevant articles. In the current paradigm of keyword-based search, each search is performed independently, placing the burden on the user to manually keep track of the relevant results of each search, e.g., by maintaining multiple open browser tabs. While a user can narrow their search by using boolean operators in their query, studies suggest that less than 5% of users use these advanced features (Spink et al., 2001). CiteNet supports an iterative search process natively and intuitively by allowing the user to refine a search with the articles from the current search results, successively narrowing the search until the majority of results are relevant to their original query.

To explore the effect of iterative searching, we build on our first case study (see §3.1) by refining the search with a second seed paper. Specifically, we chose (Giorgi and Bader, 2018), which

Table 2: Titles of the top 5 most similar articles (descending order) for the given seed(s). Bold, seed paper(s).

	PMID	Text
Seeds	28881963	**Deep Learning With Word Embeddings Improves Biomedical Named Entity Recognition**
	29868832	**Transfer Learning for Biomedical Named Entity Recognition With Neural Networks**
	31243432	HUNER: Improving Biomedical NER With Pretraining
	31501885	BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining
	31218364	Towards Reliable Named Entity Recognition in the Biomedical Domain
	29718118	D3NER: Biomedical Named Entity Recognition Using CRF-biLSTM Improved With Fine-Tuned Embeddings of Various Linguistic Information
	27283952	TaggerOne: Joint Named Entity Recognition and Normalization With semi-Markov Models

extends the contributions of the first seed paper (Habibi et al., 2017) by pre-training the model (often referred to as transfer learning). The additional seed paper implicitly narrows the search from deep learning-based methods for biomedical NER to those that exploit pre-training. The shared concepts between these seeds is reflected in the refined search results, where the top three most relevant papers (Weber et al., 2019; Devlin et al., 2018; Giorgi and Bader, 2019) each present a pre-training strategy for biomedical NER.

4 System Design

CiteNet ensures fast keyword querying of papers as well as rapid access to citation and semantic edges during the search phase by storing article information in a local Elasticsearch[3] database. This is an alternative to online querying of remote resources which may be prone to data limits, outages or unexpected changes to the API or data format.

Currently, CiteNet indexes articles from PubMed. Article metadata is batch downloaded from the U.S. National Library of Medicine[4] (NLM), parsed, formatted and quality controlled – resulting in an index size of approximately 23 million articles. Article abstracts are passed to a document encoder which learns salient feature vector representations for each article (see §4.1). A graph of semantically similar articles is then derived from these features. Citation edges are obtained from the iCite database (Hutchins et al., 2019).

The backend server and search algorithm are implemented in Node.js[5] and the front-end user interface is implemented using the React framework[6]. An overview of the CiteNet system design is given in Figure 2.

4.1 Document Embedding

To encode semantic information into the search algorithm, CiteNet uses a novel document representation method. Following other distributed document representation methods such as Doc2Vec (Le and Mikolov, 2014), the underlying model is trained in an unsupervised fashion to encode each document as a dense, low-dimensional vector with the semantic meaning of the document distributed along the dimensions. More specifically, our method is inspired by previous attempts to use autoencoders for document representation (Zhai and Zhang, 2016; Chen and Zaki, 2017). However, instead of representing documents as normalized word count vectors and training the model to re-create them, we train the model to reconstruct the input text directly, forcing it to account for word order. We hypothesize that this inherently more difficult learning objective will induce better document representations. Additionally, we incorporate a language model pre-trained on billions of words, exploiting the rich syntactic and semantic information captured by its contextual word embeddings.

We frame the model as an encoder-decoder (or seq2seq) model, and abstract away the idea of an *encoder*, *pooler*, and *decoder*. The encoder is trained to map the tokens of the input text (in our case, abstracts of scientific papers) to dense, low-dimensional vectors. The pooler is responsible for mapping these word embeddings to a single document embedding. Finally, the decoder, using the document embedding as input, attempts to re-create the input text. The entire model is trained to minimize the reconstruction loss of the input documents in an end-to-end fashion. Once training is com-

[3] https://www.elastic.co/products/elasticsearch
[4] https://www.nlm.nih.gov/
[5] https://nodejs.org/en/
[6] https://reactjs.org/

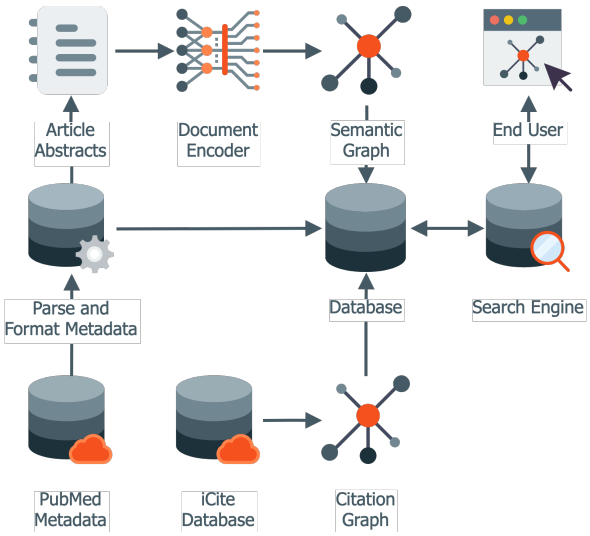

Figure 2: CiteNet system overview. All available PubMed article metadata (title, authors, abstract, journal and publication date) is downloaded from U.S. National Library of Medicine (NLM), formatted into an appropriate schema and added to CiteNet's database. Article abstracts are passed to the document encoder, which produces document embeddings. A semantic graph is then derived from these embeddings and added to the database. Citation links between articles are obtained from iCite and added to the database. Given a user query, CiteNet's search engine queries the database and produces a search result which is returned to the user as an interactive visualization. Icons made by Becris, Smashicons and Pixel perfect from www.flaticon.com.

plete, the decoder is discarded, and the encoder and pooler can be used to embed new documents. This abstraction ensures that these components remain modular and can be easily swapped and continually updated with state-of-the-art natural language processing (NLP) models. More formally, each component operates as follows:

- **Encoder**: for a given document D of n word tokens $(w_1, ..., w_n)$, maps each token to a fixed-length contextualized vector representation $\boldsymbol{Z} \in \mathbb{R}^{n \times d_{\text{word}}}$ which encode syntax and semantics.
- **Pooler**: maps the contextual word embeddings output by the encoder, $\boldsymbol{Z}$, to a single vector representation $\boldsymbol{d} \in \mathbb{R}^{d_{\text{document}}}$.
- **Decoder**: attempts to reconstruct the input document D given only the document embedding output by the pooler, $\boldsymbol{d}$.

In this work, we use a pre-trained ALBERT model as the encoder (Lan et al., 2019), where $d_{\text{word}} = 768$. ALBERT belongs to a family of pre-trained language models that have achieved state-of-the-art performance across a wide variety of NLP tasks (Dai and Le, 2015; Radford et al., 2018; Devlin et al., 2018; Howard and Ruder, 2018) and now consistently rank among the top methods for NLP leaderboards like GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019). For the pooler, we linearly project the mean of the token embeddings from the last layer of the encoder to a vector of size $d_{\text{document}} = 512$ using a feedforward neural network. We use a vanilla, one-layer transformer (Vaswani et al., 2017) as the decoder.

As a proof-of-concept, the model was trained on 50,000 PubMed abstracts centred around the case study paper, Habibi et al. (2017) (see §3). We first take all two-hop neighbours of Habibi et al. (2017) in the citation graph. In total, this constitutes 3288 articles. We then randomly sample from the set difference of the three-hop and two-hop neighbours to make up the remaining 46,712 papers. After training, we compute the cosine similarity between all pairs of document embeddings. Similarities above a stringent threshold are retained and binarized to create a semantic graph that is used in CiteNet's core search algorithm (see §4.2).

4.2 Search Algorithm

CiteNet leverages information from both a citation and semantic graph. These graphs have different topologies and capture different but complementary relationships between articles. The nodes returned by a search are scored based on their proximity to the set of seed nodes specified by the user, determined through a random walk with restart (RWR) process.

CiteNet performs several thousand RWRs on a multigraph of citation and semantic edges in order to approximate the process steady-state. Each seed node becomes the source for an equal proportion of total walks. Each walk step randomly switches between traversing citation or semantic edges in order to more fully explore the local neighbourhood of the seeds. The RWR process provides a score for each node it reaches, which corresponds to the frequency of random walk termination on that node. Nodes which have many paths to the seeds will generally be reached more often by the RWR process and be scored higher than nodes that are more isolated or not central to the seed nodes. The resulting nodes are ranked by score and the

subgraph formed by the top 30 scoring nodes (as well as seed nodes) is returned to the user.

5 Conclusion

In this paper we introduced CiteNet, a search and visualization tool for exploring scientific literature. We described the basic design elements and implementation details of CiteNet, how to use it, and presented two compelling use cases where scientific literature search is improved by a) incorporating semantic information and b) by iteratively refining a search. We believe CiteNet will be useful to the scientific community by offering an efficient and intuitive alternative to current keyword-based search engines.

Currently, we have indexed the majority of PubMed to develop a proof-of-concept. The semantic graph, however, has low coverage and our immediate aim is to expand it to cover all currently indexed articles. Moving forward, we plan to expand CiteNet to include literature from outside the biomedical field with the goal of eventually indexing all available scientific articles (as well as pre-prints) – providing a unified search engine for all disciplines. Additionally, we aim to further improve our search algorithm by a) learning document embeddings over the full article text as opposed to only abstracts and b) incorporating the citation graph topology with semantic representations directly using graph neural networks. Finally, we will continue making improvements to the user interface by introducing new visualizations, in addition to implementing user accounts, which will provide a personalized search experience.

Acknowledgments

This research was enabled in part by support provided by Compute Ontario (https://computeontario.ca/) and Compute Canada (www.computecanada.ca). We acknowledge Zain Patel, Aleksei Shkurin, Carl de Boer, Gary Bader and Bo Wang for their general comments, critiques and conceptual contributions to CiteNet's overall design, data collection and user interface. We thank Dakota St. Laurent for his technical advice on server architecture options, design and setup. Finally, we thank Matej Ušaj for his considerable help in setting up various elements of the server architecture as well as his technical advice.

References

Yu Chen and Mohammed J Zaki. 2017. Kate: K-competitive autoencoder for text. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 85–94. ACM.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

John M Giorgi and Gary D Bader. 2018. Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics*, 34(23):4087–4094.

John M Giorgi and Gary D Bader. 2019. Towards reliable named entity recognition in the biomedical domain. *Bioinformatics*, 36(1):280–286.

Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. 2017. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

B. Ian Hutchins, Kirk L. Baker, Matthew T. Davis, Mario A. Diwersy, Ehsanul Haque, Robert M. Harriman, Travis A. Hoppe, Stephen A. Leicht, Payam Meyer, and George M. Santangelo. 2019. The nih open citation collection: A public access, broad coverage resource. *PLOS Biology*, 17(10):1–6.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Amanda Spink, Dietmar Wolfram, Major BJ Jansen, and Tefko Saracevic. 2001. Searching the web: The public and their queries. *Journal of the American society for information science and technology*, 52(3):226–234.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Leon Weber, Jannes Mnchmeyer, Tim Rocktschel, Maryam Habibi, and Ulf Leser. 2019. HUNER: improving biomedical NER with pretraining. *Bioinformatics*, 36(1):295–302.

Shuangfei Zhai and Zhongfei Mark Zhang. 2016. Semisupervised autoencoder for sentiment analysis. In *Thirtieth AAAI Conference on Artificial Intelligence*.