

A Novel Content-Based Citation Recommender

Carson Craig, Spandan Garg, Andy Lee and William Zou

Introduction

- Finding correct papers to cite can be a challenge.
- Difficult for undergrad students/beginning researchers new to the field.
- Cons of existing research (Beel et al) :
 - Unreplicable
 - Pruned datasets
- Goal: Build a replicable citation recommender which utilizes syntactic as well as semantic information.

Example

Retrieve information and
associate with references

Constraint and variable ordering heuristics for compiling configuration problems

...the solutions to configuration problems can be compiled into a decision diagram [1]. We develop three heuristics for reducing the time and space required to do this. These heuristics are based on the distinctive clustered and hierarchical structure of the constraint graphs of configuration problems [2]....

[1] Defining and Evaluating Heuristics for the Compilation of Constraint Networks

[2] An improved constraint ordering heuristics for compiling configuration problems

Problem

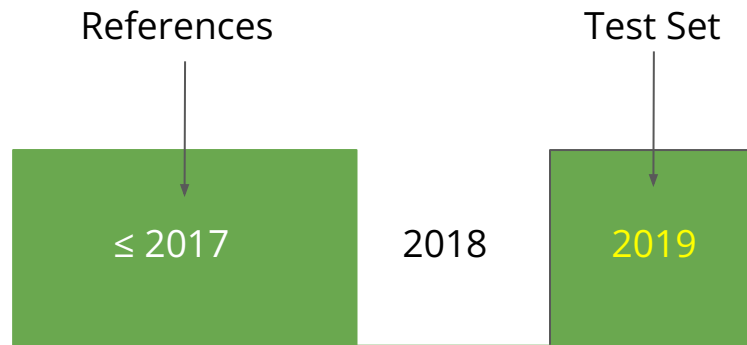
- Given a query Q and dataset D , predict the top K citations.
- Maximize the number of ground truth citations in top K results i.e. $\text{recall}@K$.
- K needs to be small because studies show 91% of people don't go past first page.

Dataset

- ArnetMiner (AMiner) dataset (Tang et al.)
- Meant for data-mining operations; has a dense citation network.
- >4 million papers
- Fields:
 - Title
 - Author
 - Abstract
 - Field Of Study
 - References
 - Year Published

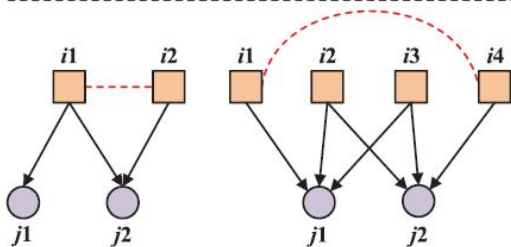
How We Perform Evaluation

- Randomly pick 100 papers from 2019.
- Remove all 2018 papers from dataset.
 - Papers published in 2019 may have been submitted in 2018.
- Measure recall@50.



Context-Based Collaborative Filtering

- Context-Based Collaborative Filtering (Liu et al.)
 - Implementation gives recall of ~22.48% on partial test set.
 - Unacceptable performance, and submitting many references is unrealistic.



(a)

(b)

Contingency Table

	$paper_{i2}$	$\neg paper_{i2}$
$paper_{i1}$	N_{11}	N_{12}
$\neg paper_{i1}$	N_{21}	N_{22}

Association Matrix

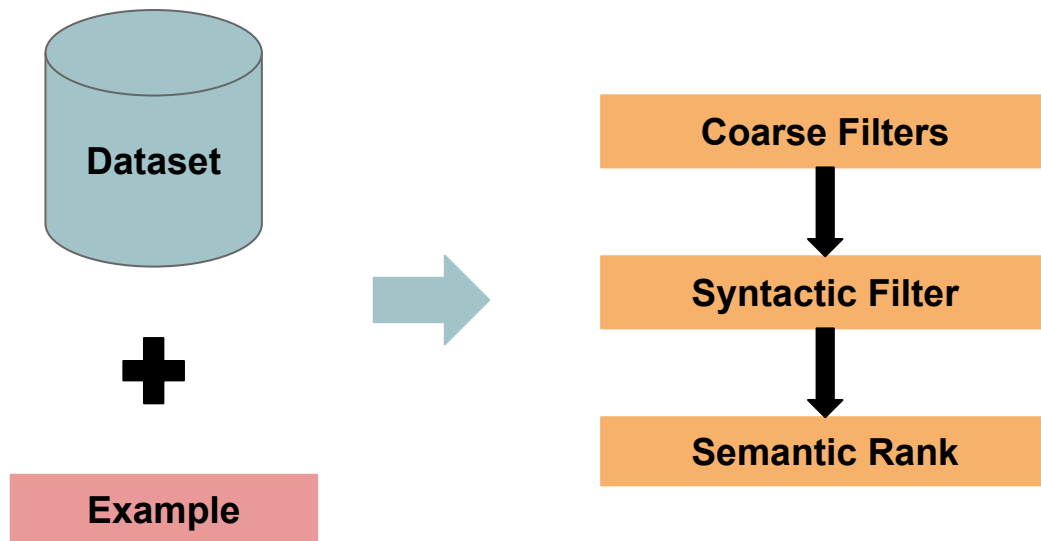
	i_1	i_2	i_3	i_4	i_5
i_1	0	1	1	0	1
i_2	1	0	1	1	1
i_3	1	1	0	1	1
i_4	0	1	1	0	0
i_5	1	1	1	0	0

- Failed attempt but gave us some valuable lessons (author distance, etc.)

Content-Based Citation Recommendation

- Content-Based Citation Recommendation (Bhagvatula et al.)
 - Nearest Neighbors on text features for candidate selection.
 - Re-rank using NeuralNet.
- Our approach draws inspiration from this.

Our Approach

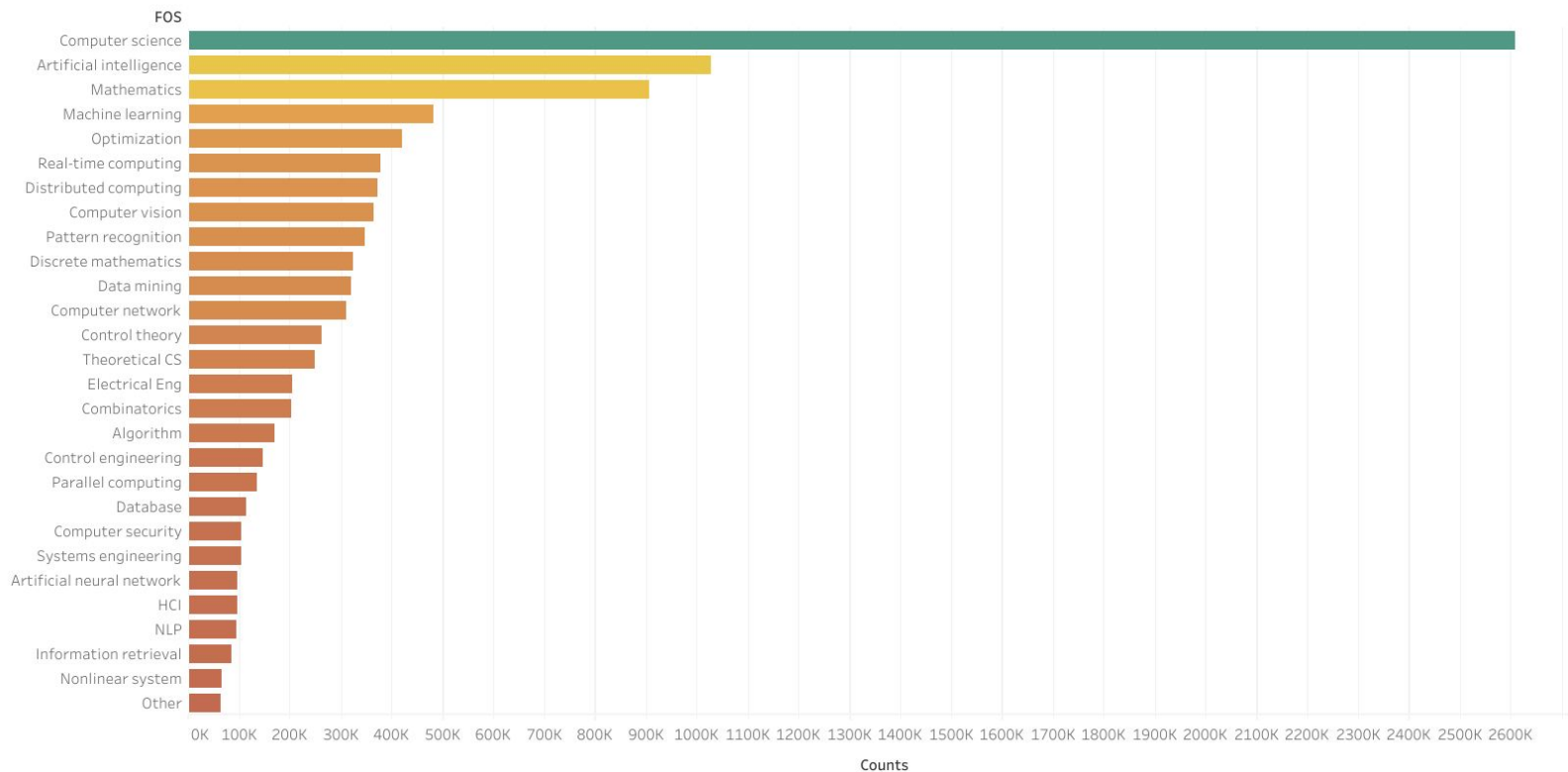


Coarse Filtering

- Goal:
 - 4 million \rightarrow <100k
 - Incur minimal penalty on max possible recall.
- Based on dataset fields (FOS, Authors, Language, Year, etc.).

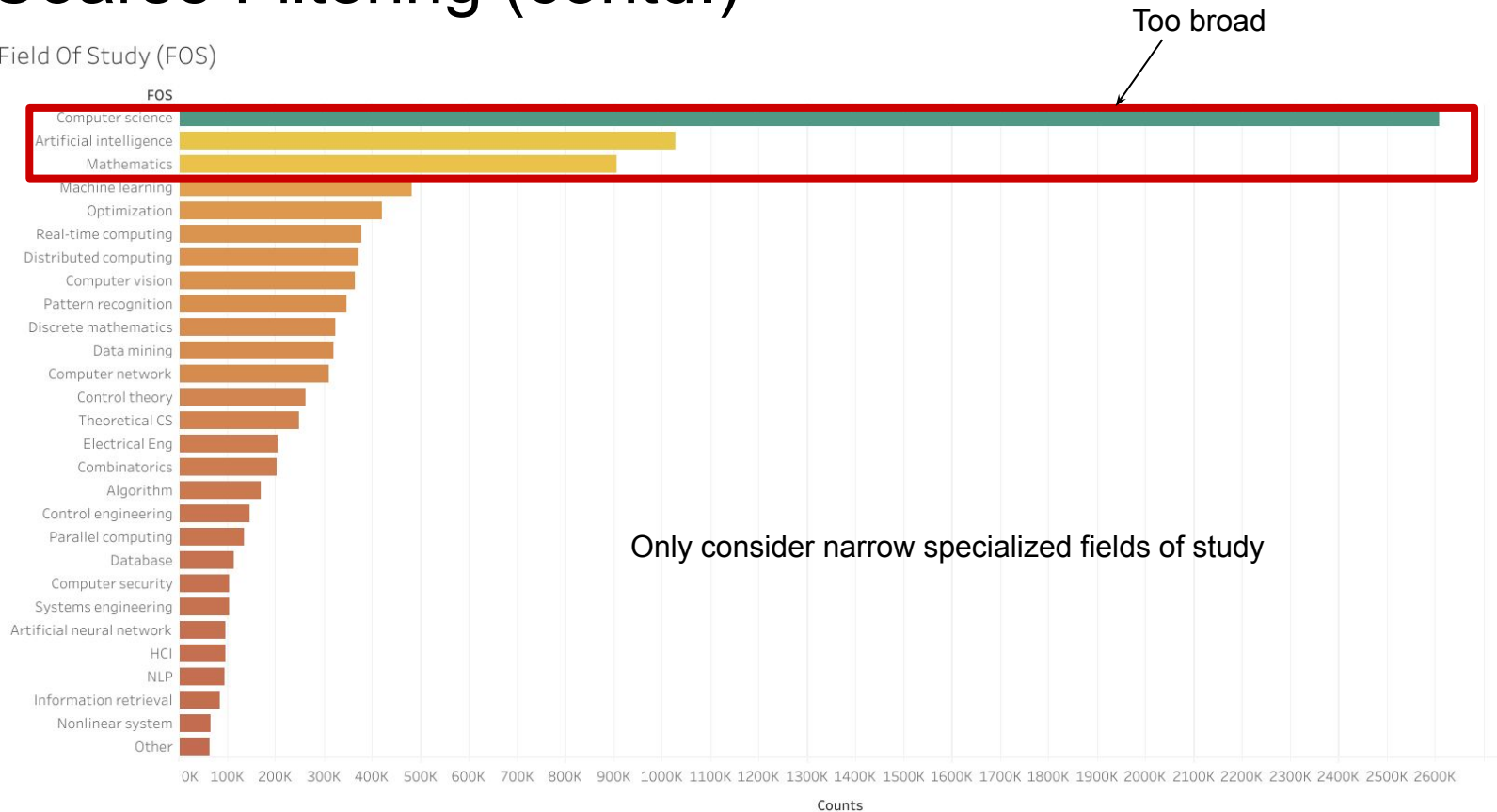
Coarse Filtering (contd.)

Field Of Study (FOS)



Coarse Filtering (contd.)

Field Of Study (FOS)



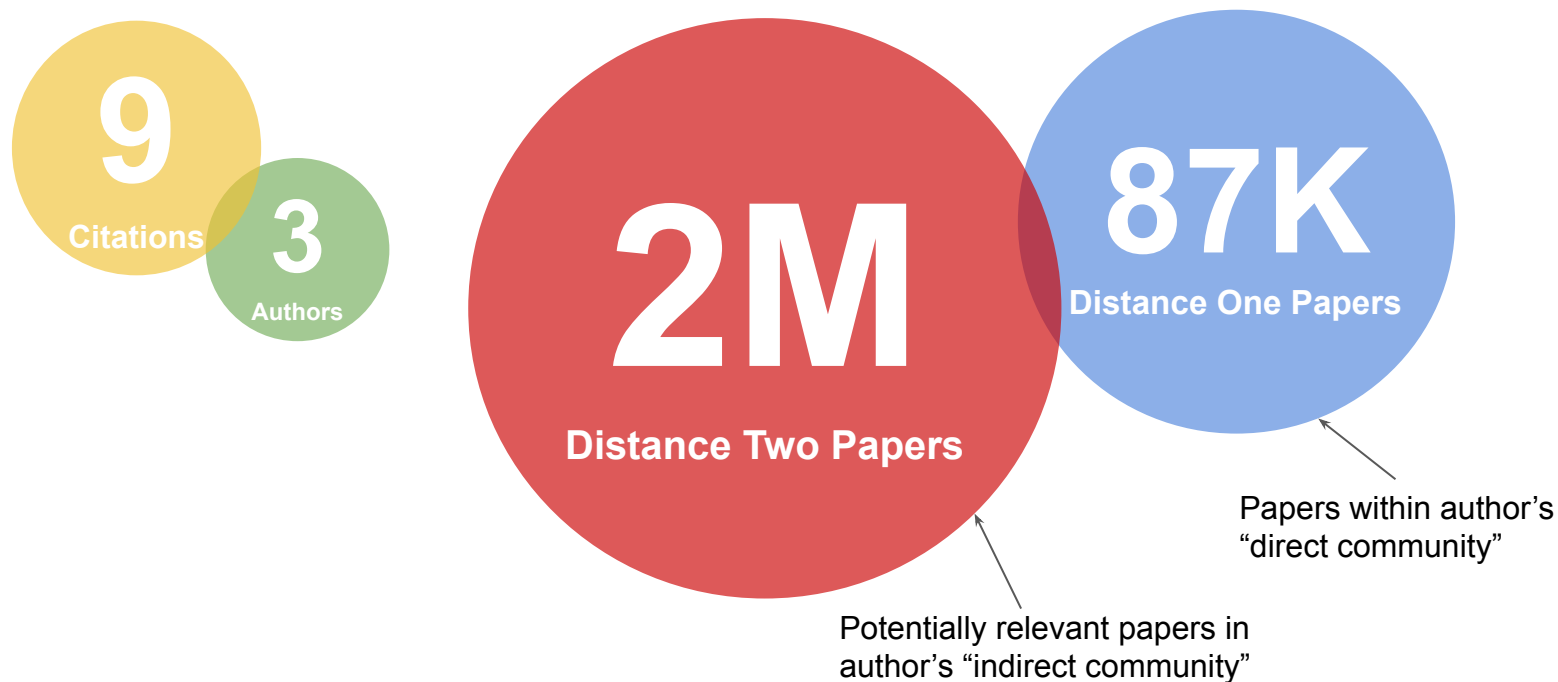
Coarse Filtering (contd.)

References & Author



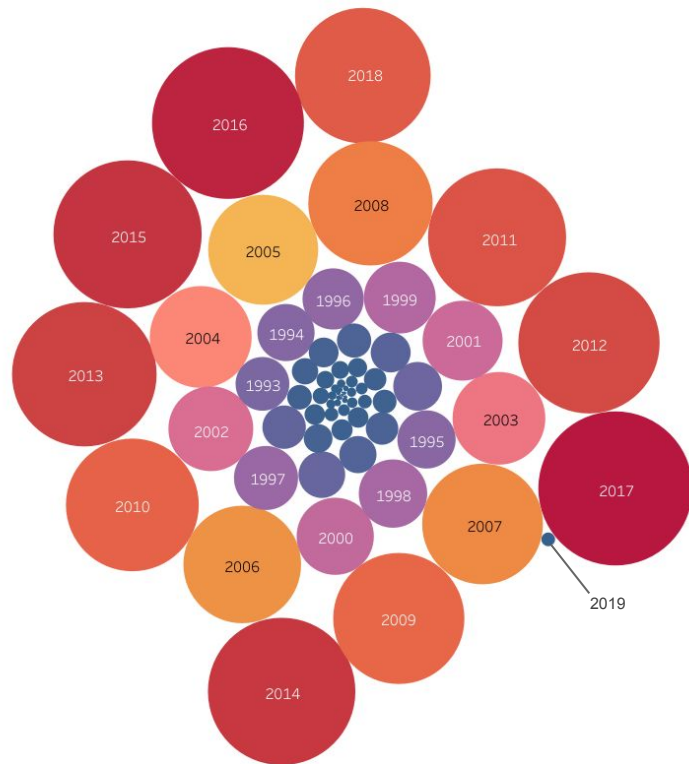
Coarse Filtering (contd.)

References & Author

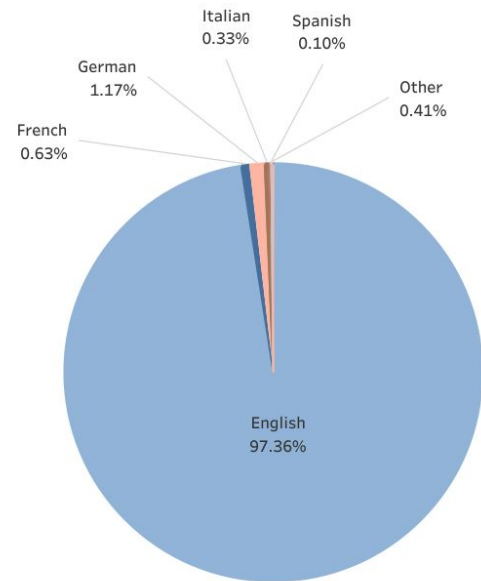


Coarse Filtering (contd.)

Year



Language



Not very helpful :(

Coarse Filtering (contd.)

- Recall after each filter:

Filter	Max Achievable Recall	Candidate Set Size
Field Of Study (FOS)	82%	500k
Distance-1 Papers	55%	80k
Distance-2 Papers	82%	1.8M
FOS + Distance-2 Papers	65%	300k
FOS + Distance-1 Papers	54%	60k

Syntactic Filtering

- Extract keywords from abstract + titles using tf-idf with threshold (~ 0.1).
- E.g.

The use of **UML** diagrams to specify a system is a well-known practice among software engineers ... The UML **metamodel**, which is contained within the UML **specification** ... **multiplicities** but it is not prescribed how to denote the multiplicities neither how to interpret them (i.e., its semantics). We propose a notation to specify multiplicities in SDs at classifier level of abstraction as well as an interpretation based on a UML metamodel extension.



uml, multiplicity, metamodel, specification, ...

- Select candidates with at least one keyword in common with the target paper.

Syntactic Filtering (Contd.)

- Combining with field filters:

Filter	Max Achievable Recall	Candidate Set Size
Syntactic Filter	84%	650k
Coarse Filters (w/ Distance-1 Authors)	54%	60k
Coarse Filters (w/ Distance-2 Authors)	65%	300k
Syntactic Filter + Coarse Filters (D2)	57%	90k
Syntactic Filter + Coarse Filters (D1)	48%	20k

- 4 million → ~20k !

Semantic Ranking

- Word Embeddings (FastText, Word2Vec)
 - Using abstracts as corpus.
- Represent each abstract as tf-idf weighted embedding.
 - Based on Neural Code Search (Sachdev et al.) by Facebook.

$$v_{paper} = \frac{1}{|paper|} \sum_{word \in paper} v_{word} * tfidf(word)$$


- Each candidate's score is cosine similarity with query paper

$$Score(q, c) = \frac{v_q \cdot v_c}{\|v_q\| \|v_c\|},$$

where q is query paper and $c \in Candidates(q)$

Semantic Ranking (contd.)

- Tried both kinds of embeddings but word2vec gives better results.



Embedding Dimension	Recall@50	Recall@100
50	2.6%	6.1%
100	7.2%	10.8%
150	8.1%	11.6%
150 (D2)	4.3%	5.9%

Query: NLM based magnetic resonance image denoising 🔍

Denoising magnetic resonance images using collaborative **NLM**

Improving Undersampled **MRI** Reconstruction Using **NLM**

Approach for **image denoising** using Zernike moments-based **NLM**

- Recall@K goes up as we increase embedding dimension.

Semantic Ranking (Contd.)

- Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al.)
 - Use next sentence prediction (NSP) task.
 - NSP gives 0 to 1 score indicating the likelihood of sentence2 following sentences1.
 - Rank by NSP score.

Model	Top 50	Top 100
BertBase	0.3%	1.2%
BertLarge	5.7%	7.8%

- Only used pretrained models without fine-tuning (future task).

Future Work

- Ways to improve semantic ranking:
 - Fine-tune BERT.
 - Try higher dimensional word embeddings.
- Ways to improve syntactic filtering:
 - Extract n-grams (e.g. 'machine learning', etc.)
- Demo website.

References

T. Soulo. 90.63% of Content Gets No Traffic From Google. And How to Be in the Other 9.37%. ahrefsblog. 2020.

<https://ahrefs.com/blog/search-traffic-study/>.

J. Tang, J. Zhang, Limin Yao, J. Li, L. Zhang, and Z. Su. ArnetMiner: Extraction and Mining of Academic Social Networks. Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2008.

<https://dl.acm.org/doi/abs/10.1145/1401890.1402008?download=true>.

H. Liu, X. Kong, X. Bai, W. Wang, T. M. Bekele, F. Xia. Context Based Collaborative Filtering for Citation Recommendation. IEEE Access. 2018.

<https://ieeexplore.ieee.org/document/7279056>.

C. Bhagavatula, S. Feldman, R. Power, and W. Ammar. Content-Based Citation Recommendation. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 2018.

<https://www.aclweb.org/anthology/N18-1022.pdf>.

J. Devlin, M. Chang, K. Lee, K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019. Proceedings of NAACL-HLT 2019. <https://arxiv.org/abs/1810.04805>.

S. Sachdev, H. Li, S. Luan, S. Kim, K. Sen, S. Chandra. Retrieval on Source Code: A Neural Code Search. Proceedings of 2nd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages. 2018. <https://people.eecs.berkeley.edu/~ksen/papers/ncs.pdf>

Thank you!

Prof. Charles Clarke

Prof. Pascal Poupart

Prof. Patrick Lam

Prof. Derek Rayside

End