Investigating the Overlap and Divergence in Search Results between Annoy and Lucene

CSC 583
Bhavya Sharma, Elliot Justice, Urvika Gola

04 May 2023

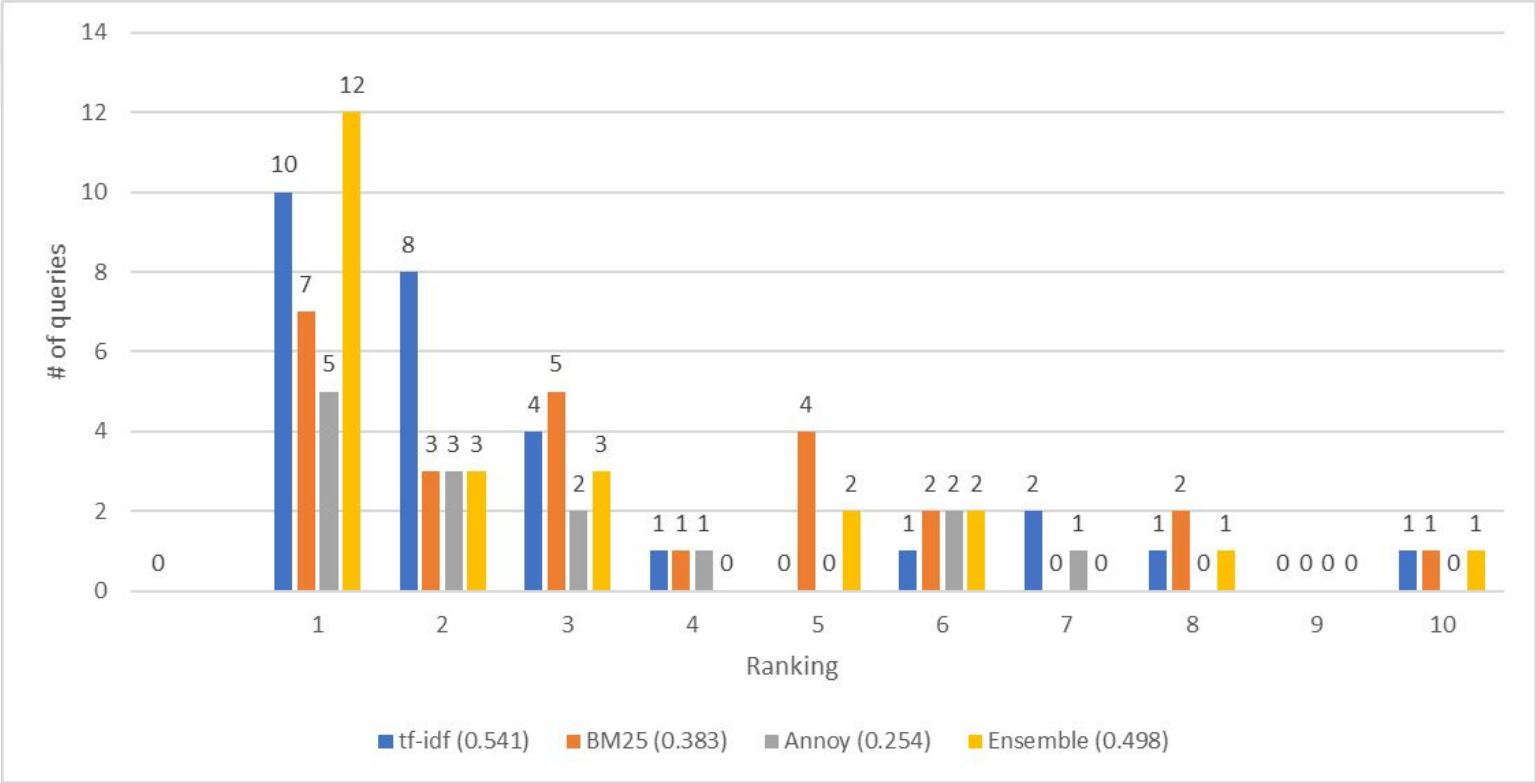
Project Background

- Companies regularly need to answer or find additional information on domain specific questions
- Commonly assume that classic search too limiting and may not aid in information discovery
- Search mechanisms built into products are not easily tunable to specific tasks
    - AI models may not be generalizable to domain specific due to overfitting on training data
    - May not be able to expose internal data to 3rd party cloud search services for security/regulatory reasons
- How well would a simplistic hybrid approach combining AI and Classical search make up for the drawbacks for the individual components?

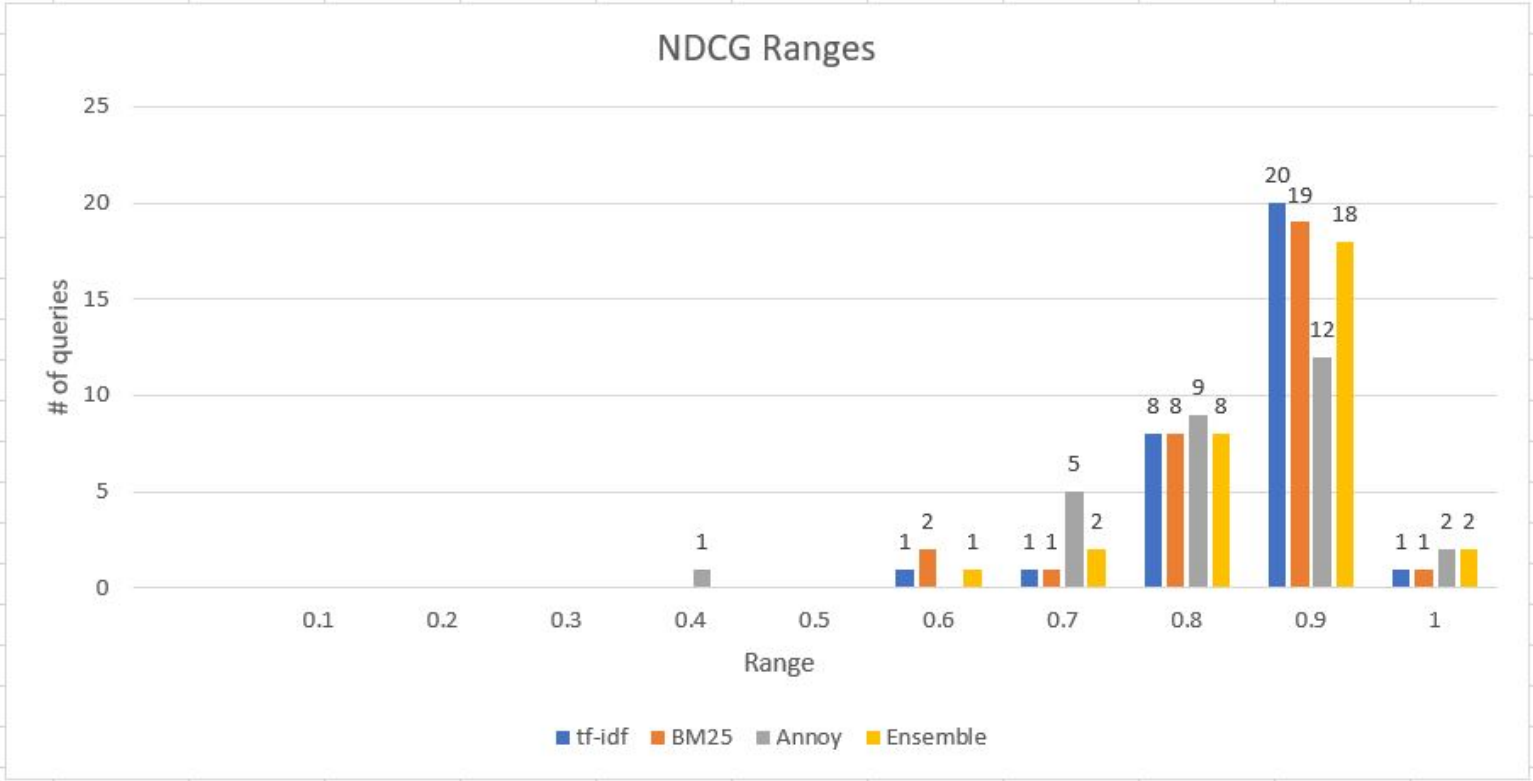
Approach and Benefits

- Using dataset of Arxiv.org Abstracts as substitute for domain specific data
- Compared Performance of Lucene along side an Transformer Model with Nearest Neighbors Indexing
- Able to index 2.2 million document in 6 hours and using minimal compute resources and storage space
- Incorporates moderate tunability without complicated Page and Link ranking methods

Performance Assessment - MRR



Performance Assessment - NDCG



Score Averages

	Tf-idf Ranking	BM25 Ranking	Annoy Ranking	Ensemble (BM25 + Annoy)
MRR Average	0.541	0.383	0.254	0.498
NDCG Average	0.900	0.897	0.814	0.906

Conclusions

- Ensemble Scoring produced results on par with Classical Search methods
- Sentence Transformer + Annoy produced lackluster results on its own
- Many opportunities available to tune Annoy to further increase overall performance when combined with classical search
- Hardware limitations likely biggest factor in Annoy performance

What we learned

- Project worked as a real-world proxy to building search capabilities on business specific data
- Understanding of questions to ask as part of a development team when working on a similar project
- What additional education and training may be required of larger development team on how to properly test and validate a search feature in an actual product
- Comment: AWS and Azure ElasticSearch are both built using Lucene