

INTRODUCTION

- **Electronic discovery** (eDiscovery) refers to discovery in civil litigations that deal with the exchange of information in electronic format.
- The corpus may have documents, PDFs, e-mails, audio or video files and the size of the data that needs to be searched is large.

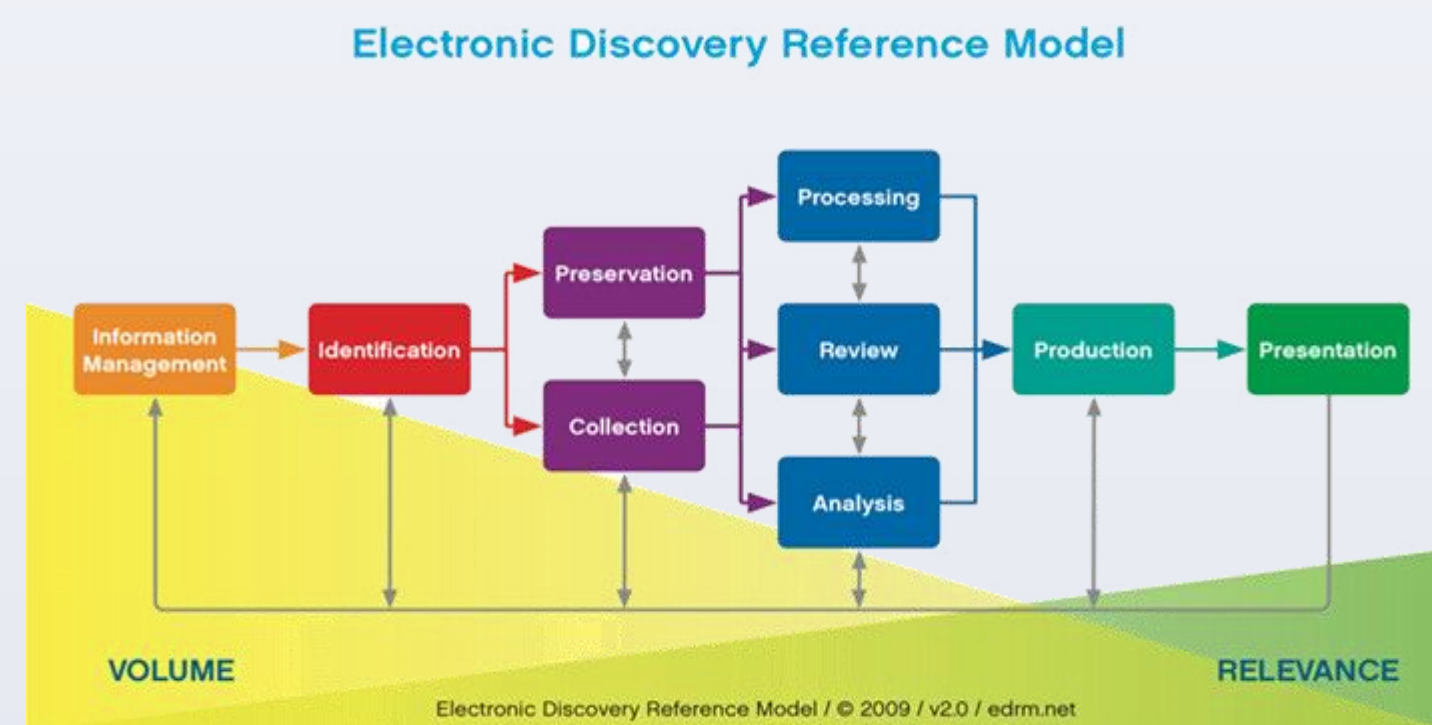


Figure 1 - eDiscovery Reference Model

SYSTEM OVERVIEW

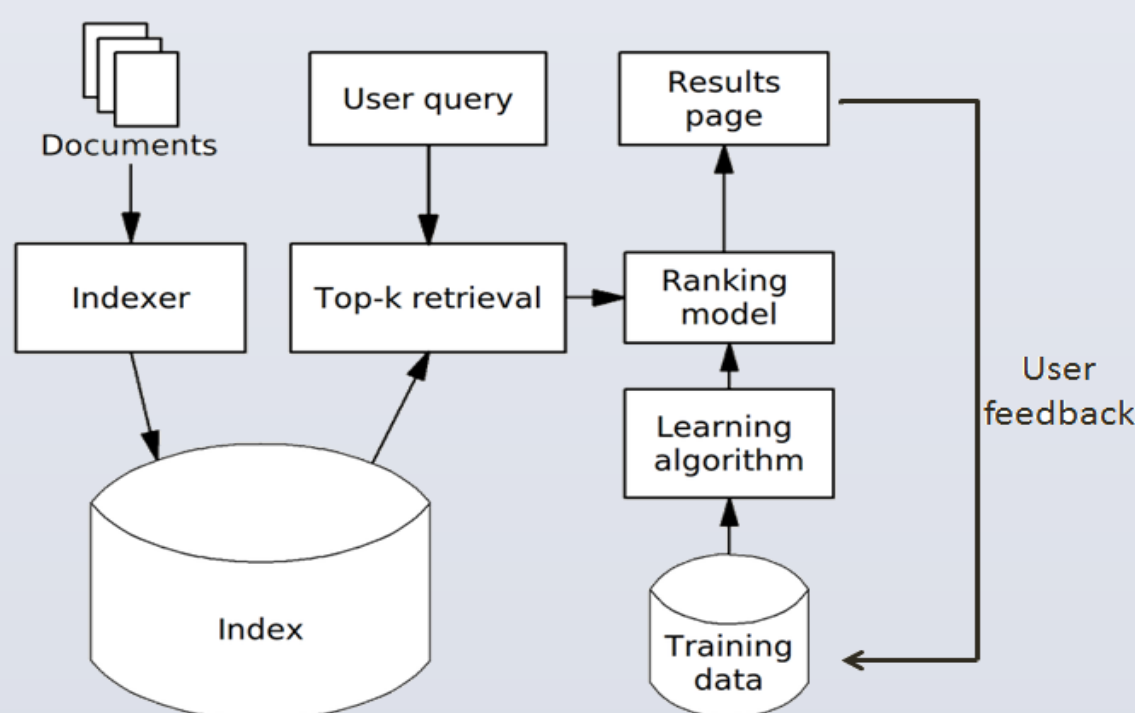


Figure 2 – Overview of Metadata based Indexing and Searching

- 1. Indexer:**
 - Indexes the set of documents based on metadata attributes.
 - This would help in the *searching* phase by retrieving documents faster.
- 2. Ranking model:**
 - Predicts the relevancy scores for the top-k retrieved documents based on the learning algorithm.
 - Retrieves the top-k documents for a query.
- 3. Learning Algorithm:**
 - Progressively learns using the relevancy scores assigned by the user.
 - Minimizes the difference between the ideal list based on the user feedback and retrieved list from top-k retrieval.
- 4. User Feedback :**
 - Captures the relevancy scores from the user
 - Feedbacks are used to learn the ranking function iteratively

METADATA BASED INDEXING

- Indexing facilitates fast and accurate information retrieval.
- Metadata based indexing facilitates exploring a corpus based on multiple facets.
- We have accomplished this using the Apache Lucene Framework - an open source information retrieval software library.

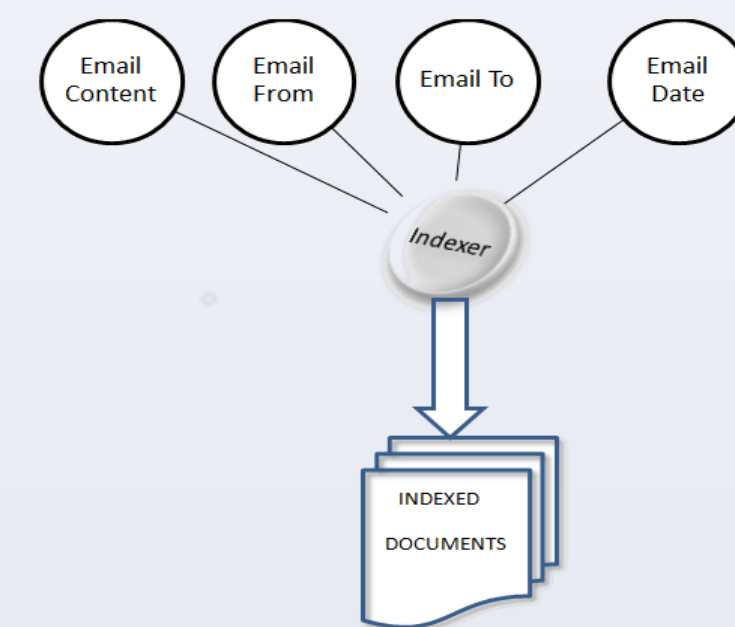


Figure 3 – Schematic representation of Metadata based indexing for an email corpus

METHODS FOR LEARNING-TO-RANK

- 1. Point-wise:**
 - Each query-document pair in the training data has an ordinal score.
 - Learning-to-rank \Leftrightarrow regression problem ($\{query, document\} \Rightarrow score\ prediction$)
- 2. Pair-wise:**
 - Learning a binary classifier that can tell which document is better in a given pair of documents.
 - Learning-to-rank \Leftrightarrow classification problem
 - Attempts to minimize the avg. no. of inversions in ranking.
- 3. List-wise:**
 - These algorithms try to directly optimize the value of one of the above evaluation measures, averaged over all queries in the training data.

The algorithms we used in our experimental analysis are:

- 1. AdaRank:**
 - Repeatedly constructs weak rankers (learners) on the basis of re-weighted training data.
 - Linearly combines the weak rankers for making ranking predictions.
- 2. ListNet:**
 - Has a probabilistic method to calculate the listwise loss function.
 - Transforms both the scores of the documents assigned by a ranking function and the judgments of the documents given by humans into probability distributions.
 - Uses learning to rank method with a listwise loss function, a Neural Network as model and Gradient Descent as the algorithm.

3. RankBoost:

- Employs a boosting based technique similar to AdaRank. However it is a pairwise boosting algorithm.

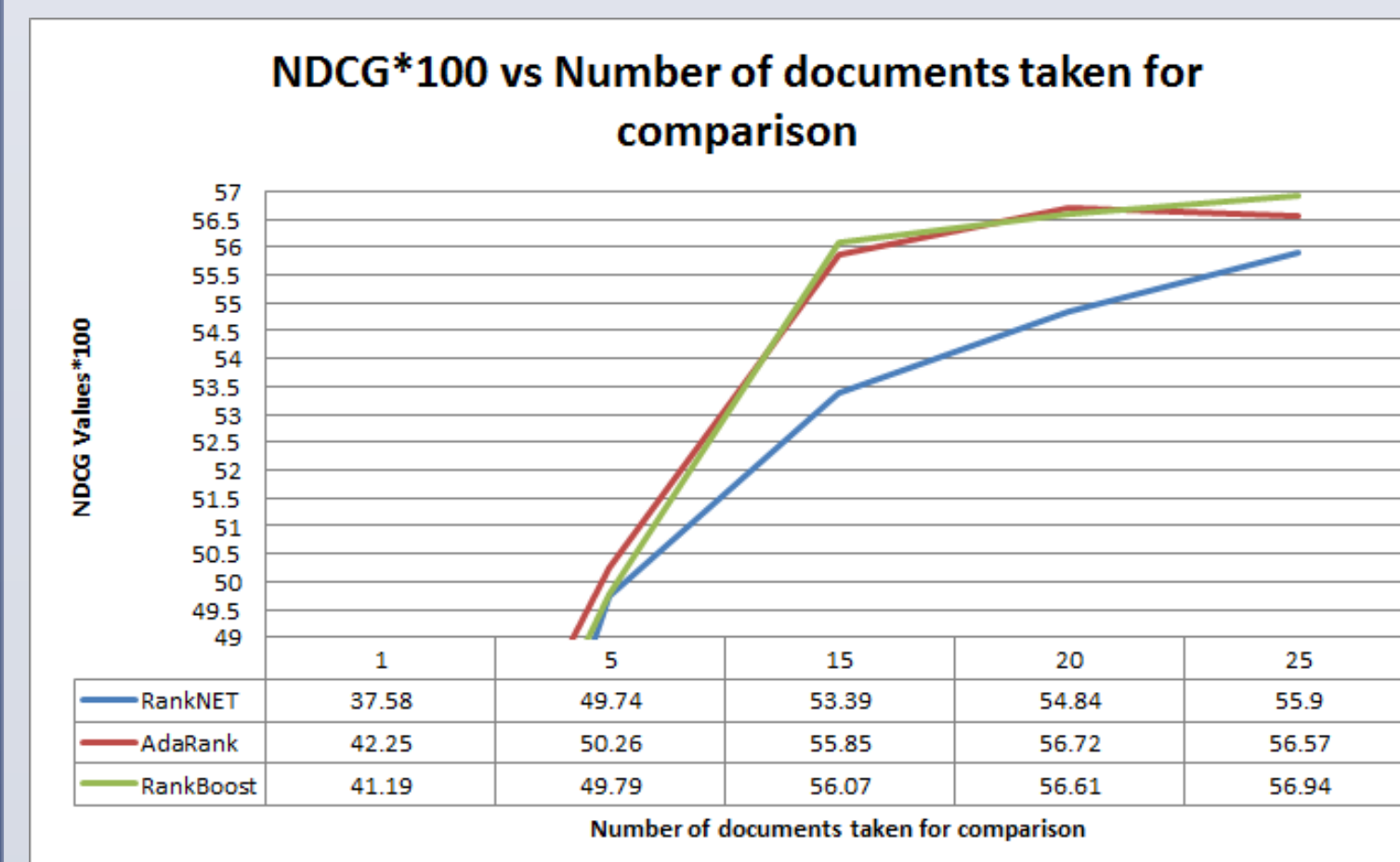
Some useful features for learning to rank algorithms are TF, TF-IDF, BM25.

RESULTS

Comparing learning-to-rank algorithms:

- Shows the comparison between listwise and pairwise ranking algorithms in terms of their performance for Normalized Discounted Cumulative Gain (NDCG).
- Higher the value of NDCG, better is the performance of the ranking algorithm.
- Compares three algorithms – AdaRank, a listwise algorithm, RankBoost and RankNet, two pairwise algorithms.

The X-axis shows the number of previous documents taken for comparison to calculate NDCG values.



Inferences from our experiments:

- Listwise ranking algorithms have better NDCG values and hence their performance in ranking is better than that of pairwise ranking algorithms.
- Pairwise and listwise algorithms perform better when a higher number of previous documents are taken into consideration while calculating NDCG values.

DISCUSSION and FUTURE WORK

- We implemented a metadata-based indexing and searching mechanism which allows efficient retrieval of relevant documents from a large corpus.
- It is a feedback-based system whose ranking model will evolve iteratively based on user feedback.
- Future work includes feature selection for learning to rank in eDiscovery and implementation of a learning to rank algorithm.

REFERENCES

1. Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. (ICML '07)
2. Taesup Moon, Alex Smola, Yi Chang, and Zhaohui Zheng. 2010. IntervalRank: isotonic regression with listwise and pairwise constraints.(WSDM '10).
3. Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. (ICML '05).
4. Freund, Yoav, et al. "An efficient boosting algorithm for combining preferences." *The Journal of machine learning research* 4 (2003): 933-969.
5. Jun Xu and Hang Li. 2007. AdaRank: a boosting algorithm for information retrieval. (SIGIR '07).

ACKNOWLEDGEMENTS

We thank Dr. Daisy Wang for giving us the opportunity to work on this project. We also thank Clint P. George for his guidance throughout the project.