Information Retrieval

V S S Anirudh Sharma

LATENT SEMANTIC ANALYSIS

AND THE BIGRAM VECTOR SPACE

Introduction

- Information Retrieval System
  - Obtain relevant word documents from the data collection
  - Rank them according to their order of relevance
  - Multiple Approaches
    - Vector Space Model using TF-IDF
    - Latent Semantic Analysis
    - Adding Bigram Terms
    - Word Pruning Techniques

Implementation

- We built a search engine from scratch in our previous assignments using the Vector Space Model (VSM). Below are the methods we implemented in the VSM search engine:
  - Sentence Segmentation of documents
  - Tokenization of the segmented sentences
  - Removal of stopwords
  - Stemming of the tokenized words
  - Inverted Index list
  - Vector representation of articles and queries
  - Finding Cosine similarity between articles and queries
  - Evaluating IR search engine using Precision, Recall, F-measure, MAP and nDCG

Problem Statement

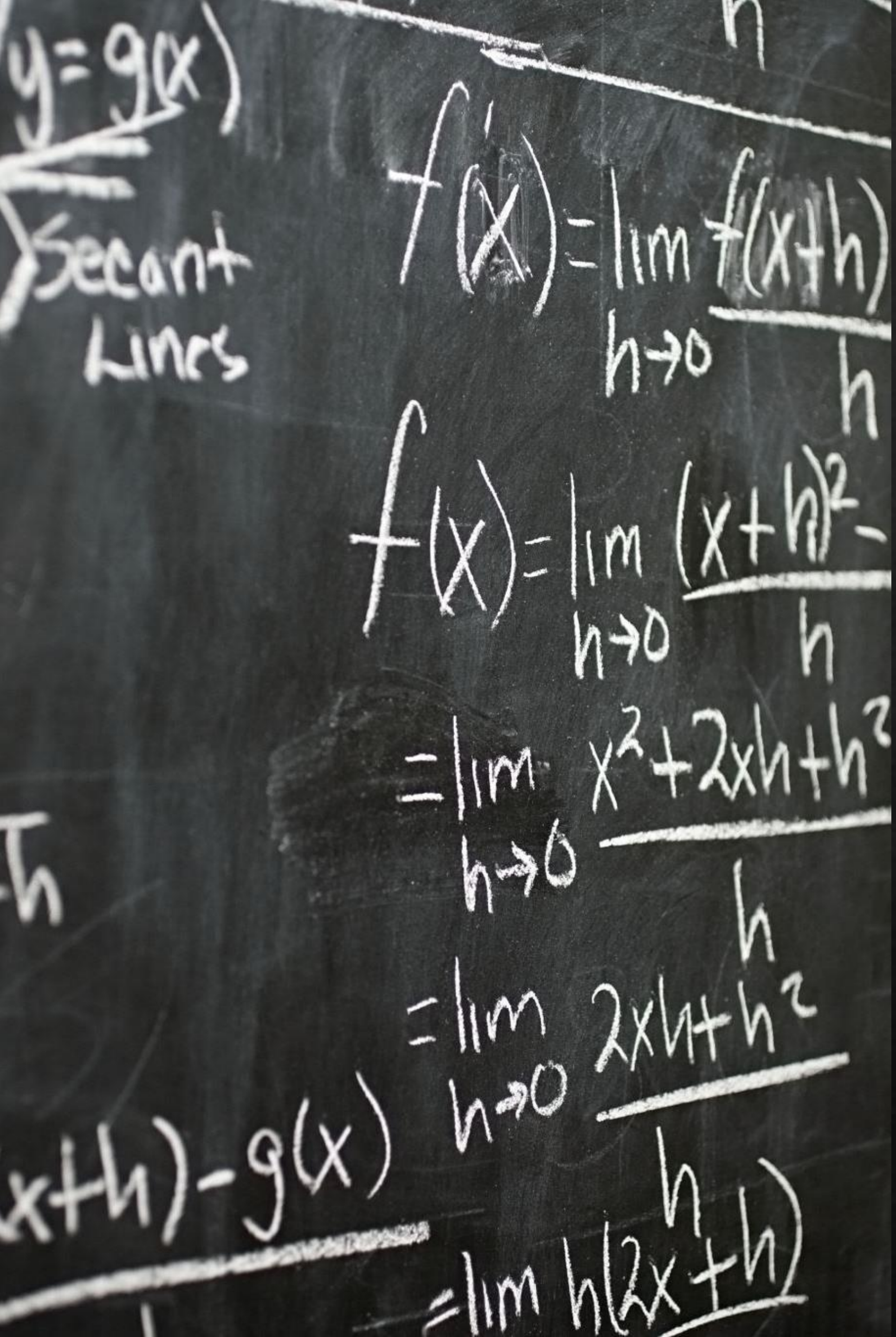
- Improve the model by:
  - Adding Semantic Relatedness among words
  - Overcome the problem of Synonymy
  - Use an external Document Corpus (like Wikipedia) as a knowledge base.
  - Preserve the order of tokens

Proposed Methodology

- Methods to improve and optimize our search engine:
  - Bigrams to model the order of tokens
  - Latent Semantic Analysis (LSA) to bring out hidden concepts
  - Using Explicit Semantic Analysis to include explicit concepts
  - And mapping their relation to overcome semantic unrelatedness

Preferred Performance Metrics

- The performance of different IR Systems is judged based on the nDCG and MAP scores.
- When the relevant documents don't have an ordering among them, MAP is used.
- nDCG is used when the relevance is ranked.
- nDCG is more appropriate for Cranfield Dataset, as the relevance score is provided.



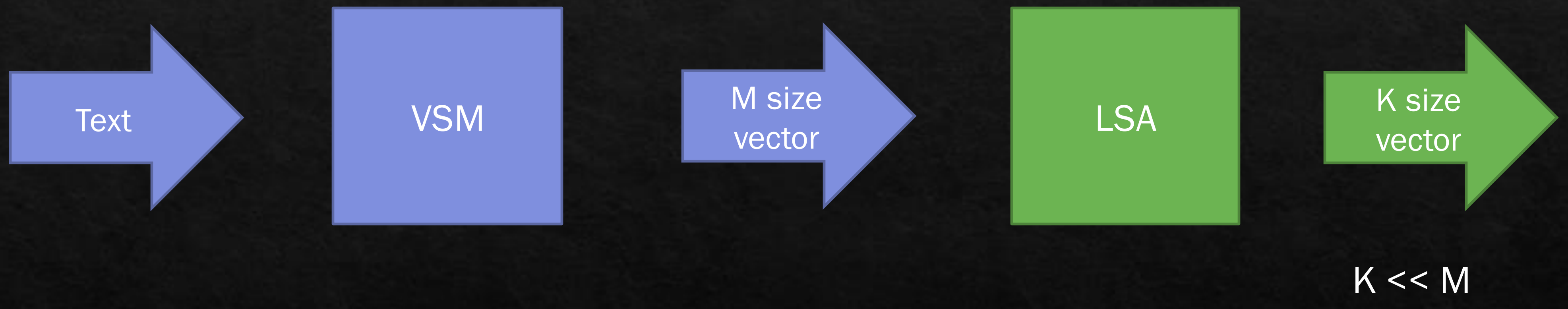
HYPOTHESES

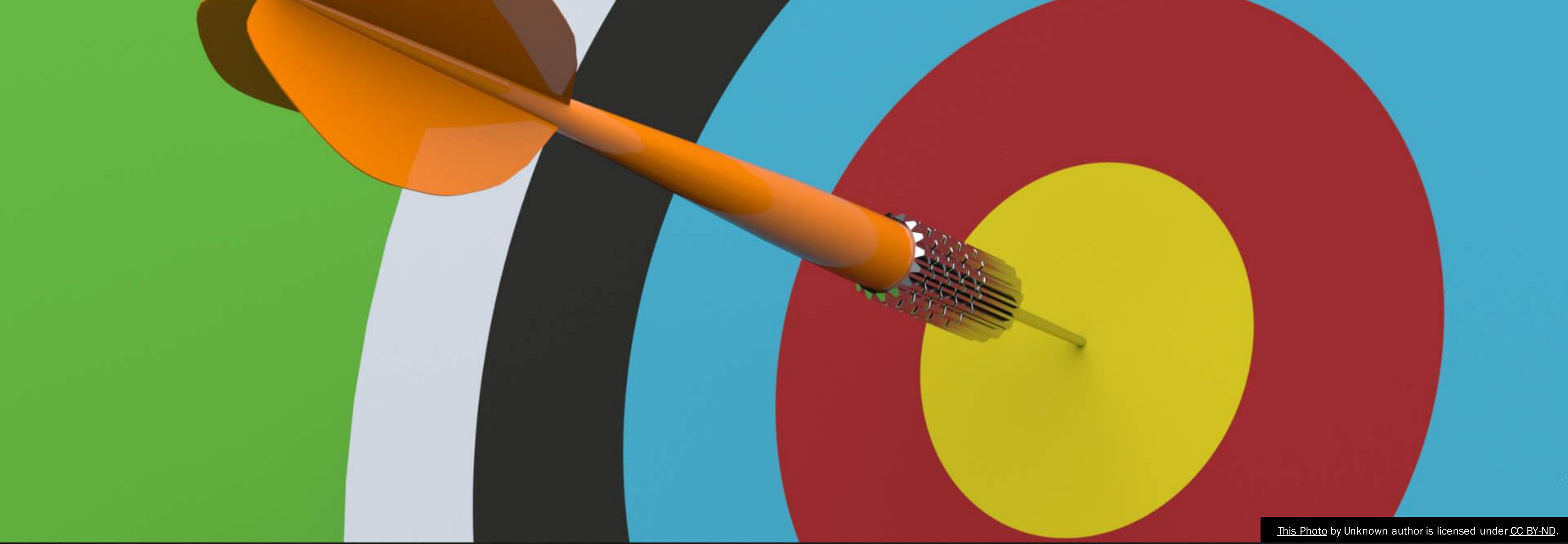
HYPOTHESES

- LATENT SEMANTIC ANALYSIS MAKES MODELS FASTER WHILE EITHER IMPROVING THE EFFECTIVENESS OF THE MODEL OR KEEPING IT THE SAME
- "BIGRAM" VECTOR MODEL WOULD CONSIDER WORD ORDER IN THE CORPUS AND GIVE BETTER RESULTS THAN ESA AND NESA FOR THE CRANSFIELD DATASET

THE LSA HYPOTHESIS

VECTOR SPACE MODEL

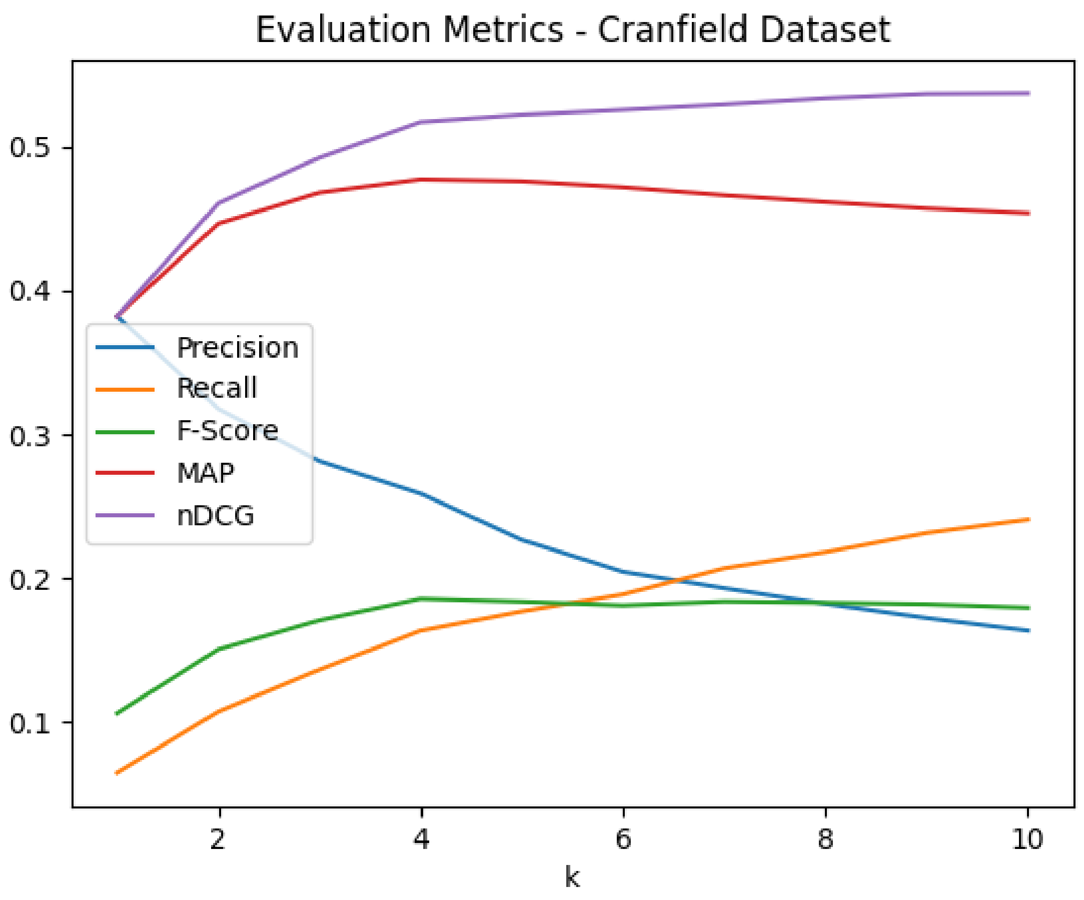




EFFECTIVENESS

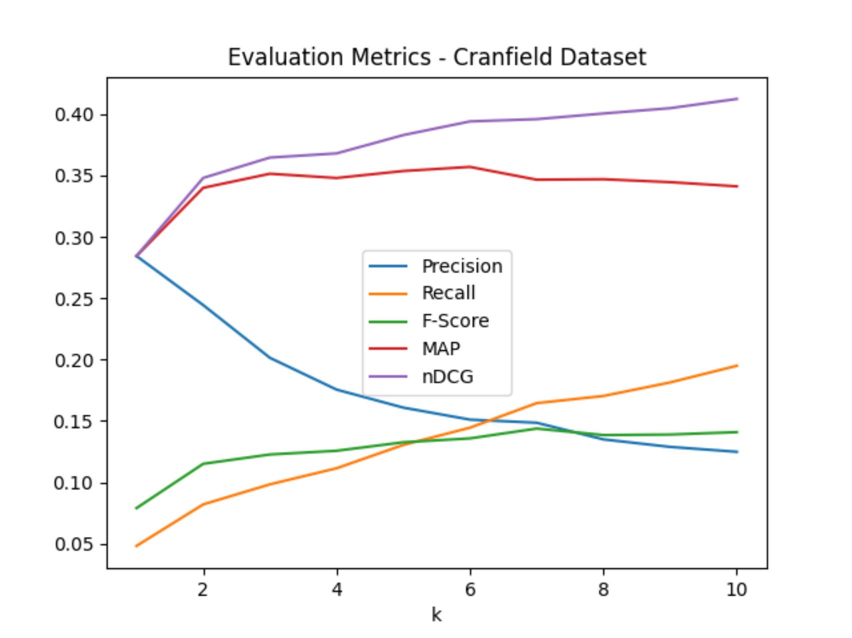
ESA

- VSM with TF-IDF vectors.
- Reduced set of words used in TF-IDF
- Using an external corpus(wikipedia) as knowledge base(ESA)
- Dimensionality Reduction using best set of concepts (LSA)

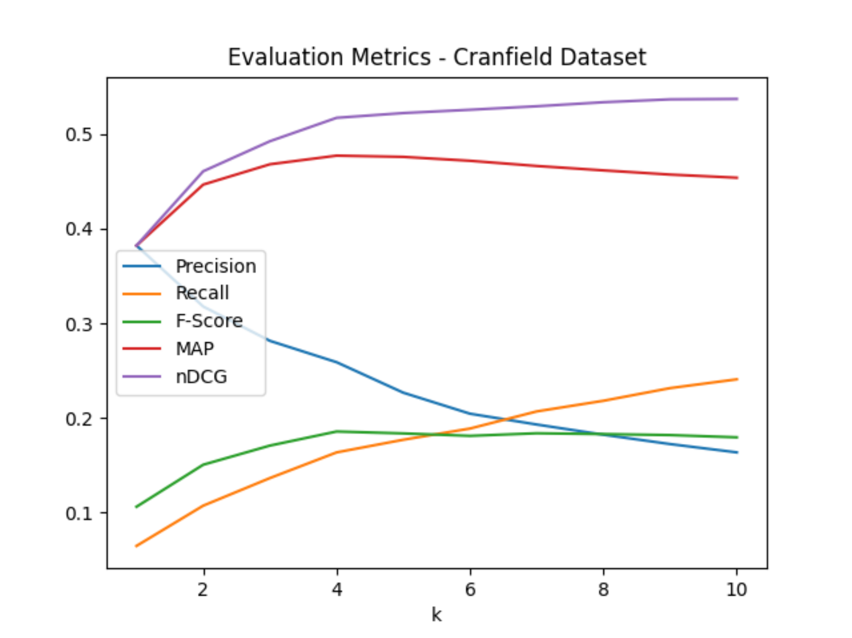


ESA BASED SEARCH

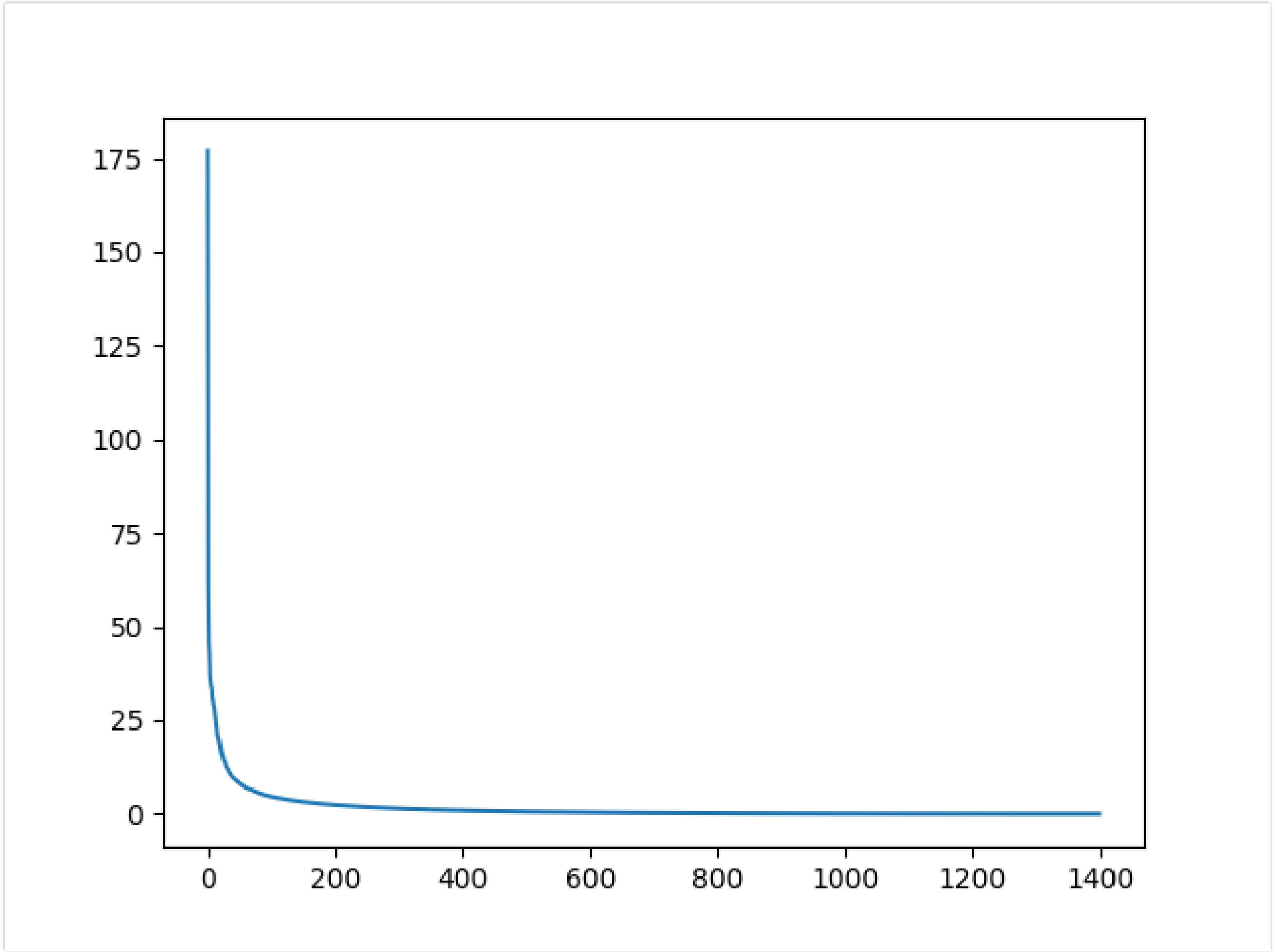
WITHOUT LSA(M=2404)



WITH LSA(350)

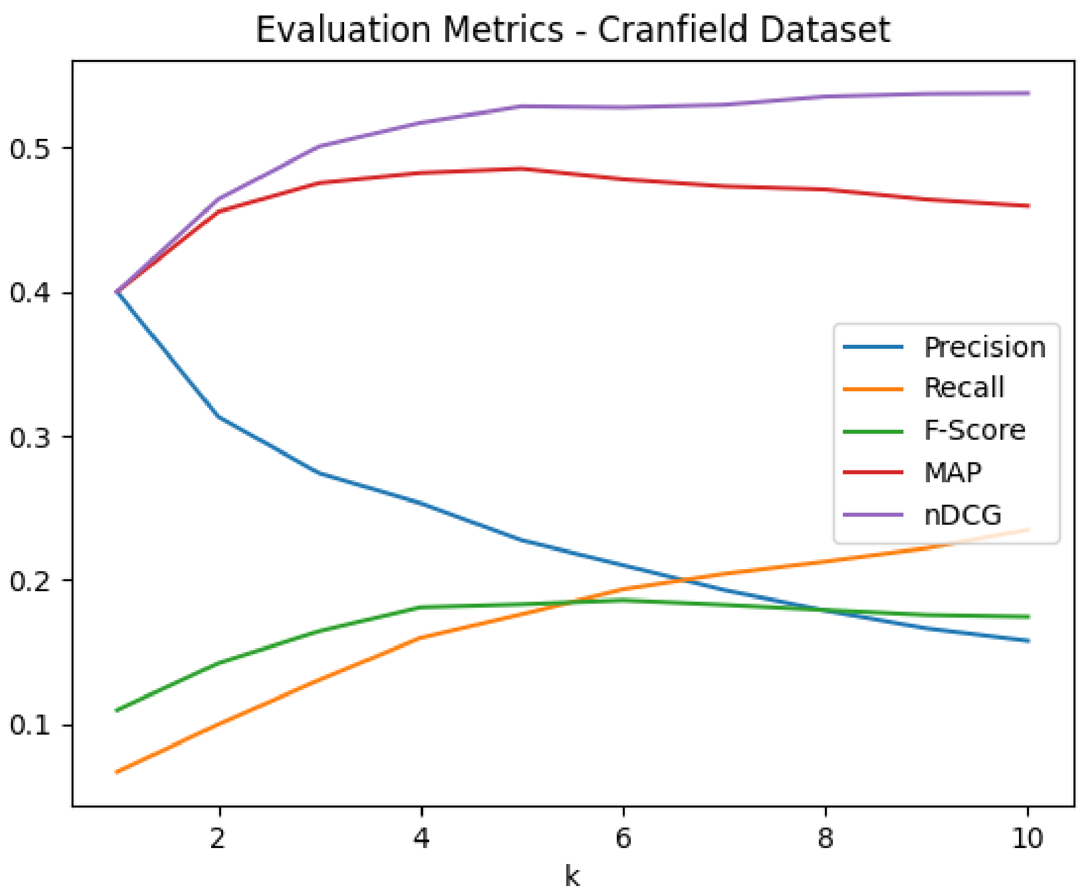


EIGEN VALUES IN LSA over ESA MODEL



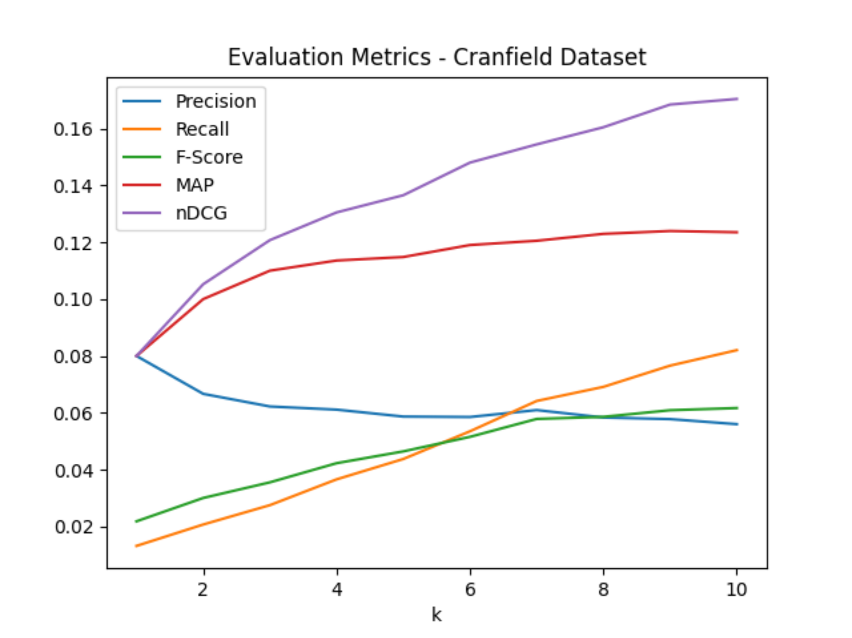
NESA

- VSM with TF-IDF vectors.
- Reduced set of words used in TF-IDF
- Using an external corpus(wikipedia) as knowledge base(ESA)
- use relatedness between the dimensions of the distributional vectors to overcome the orthogonality in ESA model (NESA)
- Dimensionality Reduction using best set of concepts (LSA)

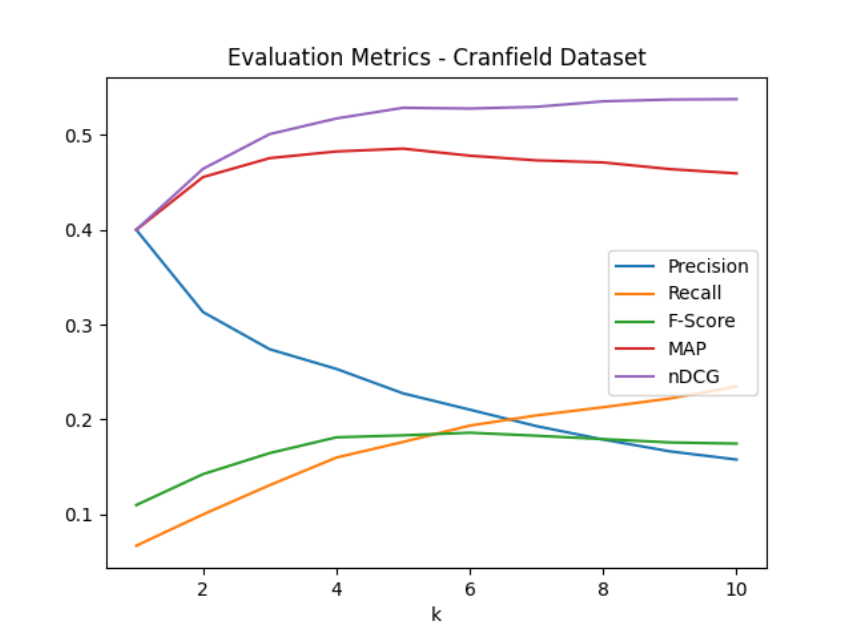


NESA based search

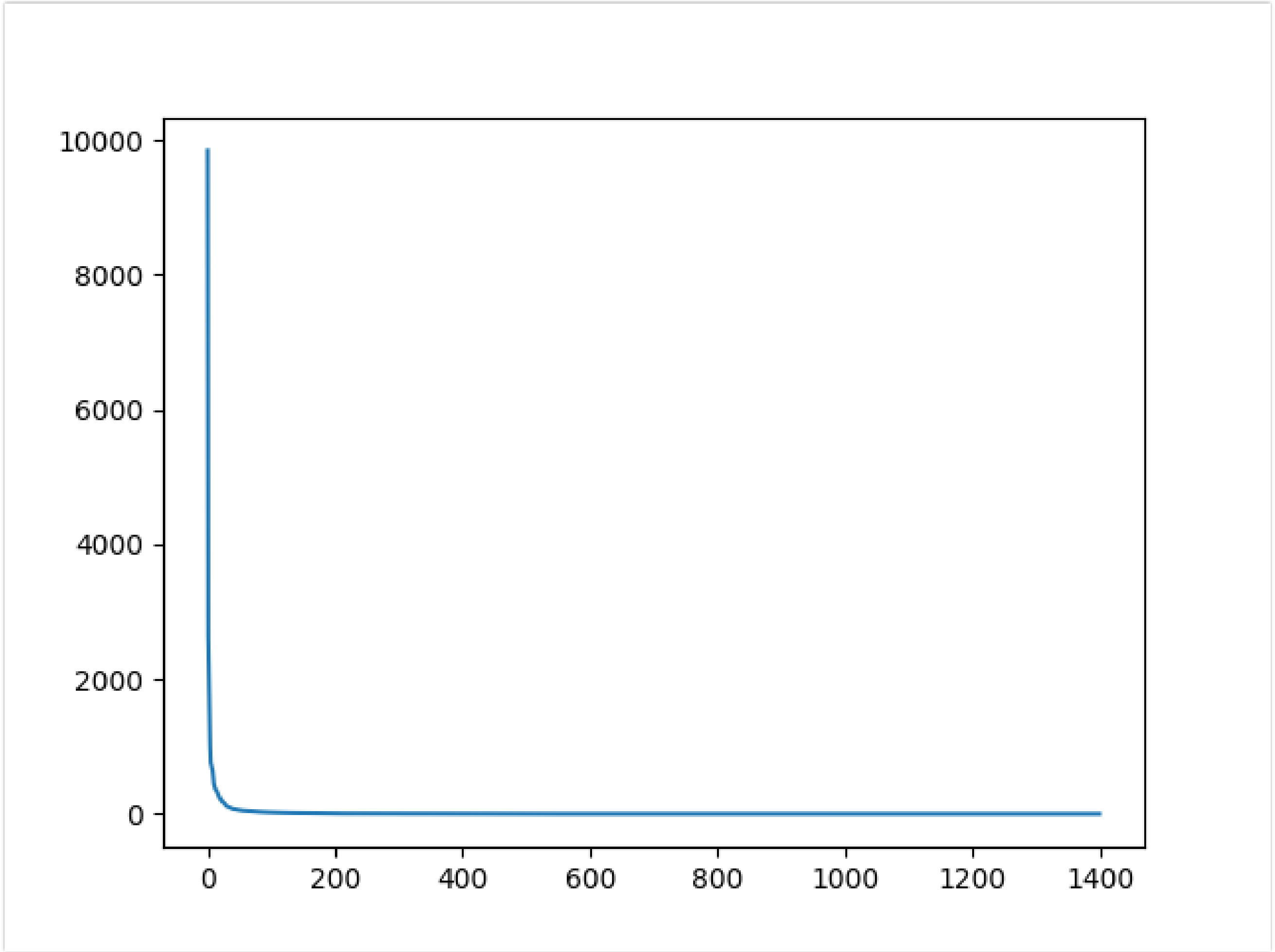
Without LSA (M=2404)



With LSA(K = 350)



EIGEN VALUES IN LSA over NESA MODEL



```
similarity laws
high speed
chemical kinetic
pressure distributions
```

THE BIGRAM VECTOR SPACE HYPOTHESIS

Bigram Vector Space

- Consider the following phrases:
  - similarity laws
  - high speed
  - chemical kinetic
  - chemical equilibrium
- We want the two words to be co-located next to each other for an efficient search

Bigram Vector Space

Unigram Vector space computes vectors for

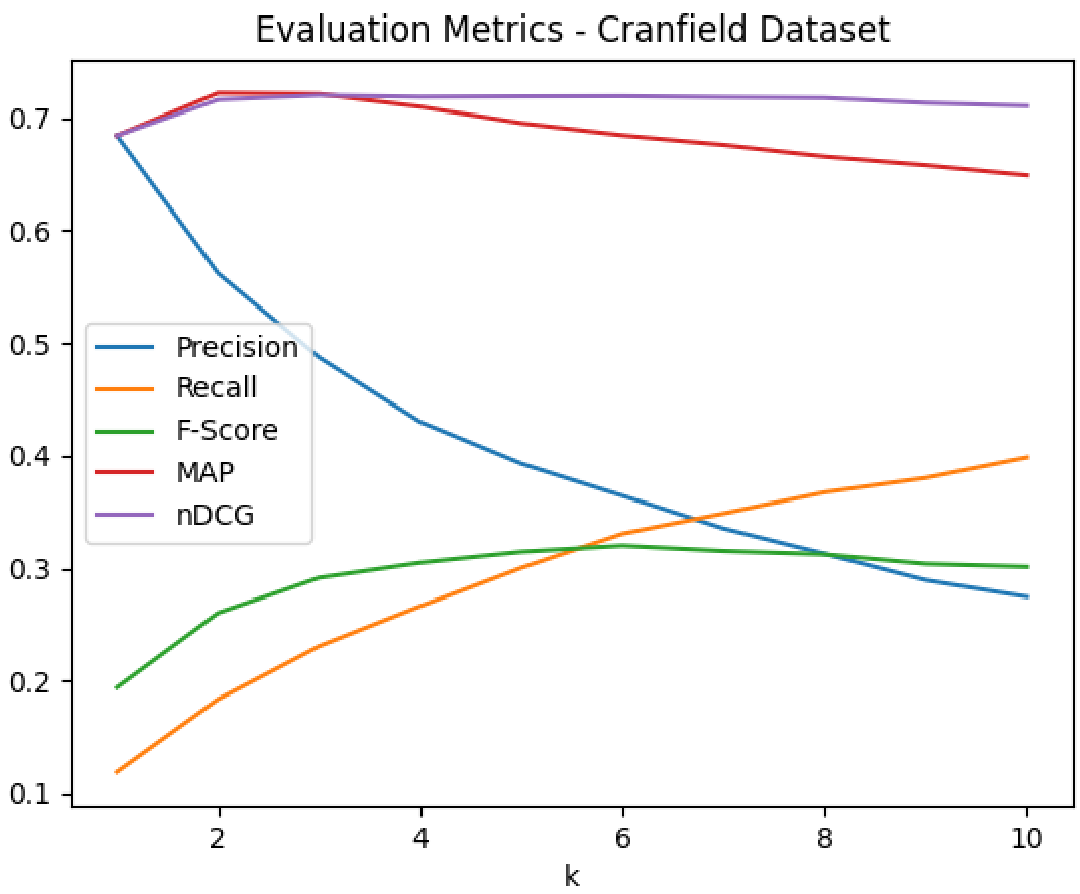
- similarity
- laws
- high
- speed
- chemical
- kinetic
- equilibrium

Bigram Vector Space

- Bigram Vector space computes vectors for
  - similarity-laws
  - High-speed
  - chemical-kinetic
  - chemical-equilibrium
- This ensures we are a considering the collocation of words too.
- Cransfield dataset has a ton of such phrases

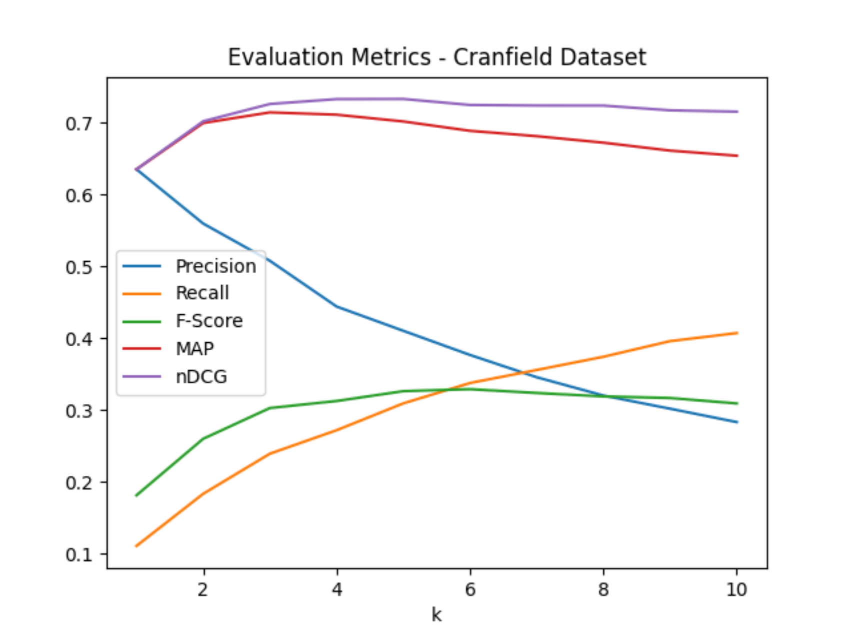
Bigrams

- Bigram VSM with TF-IDF vectors.
- Reduced set of words used in TF-IDF
- Dimensionality Reduction using best set of concepts (LSA)
- Considers the order of occurrence of terms/tokens

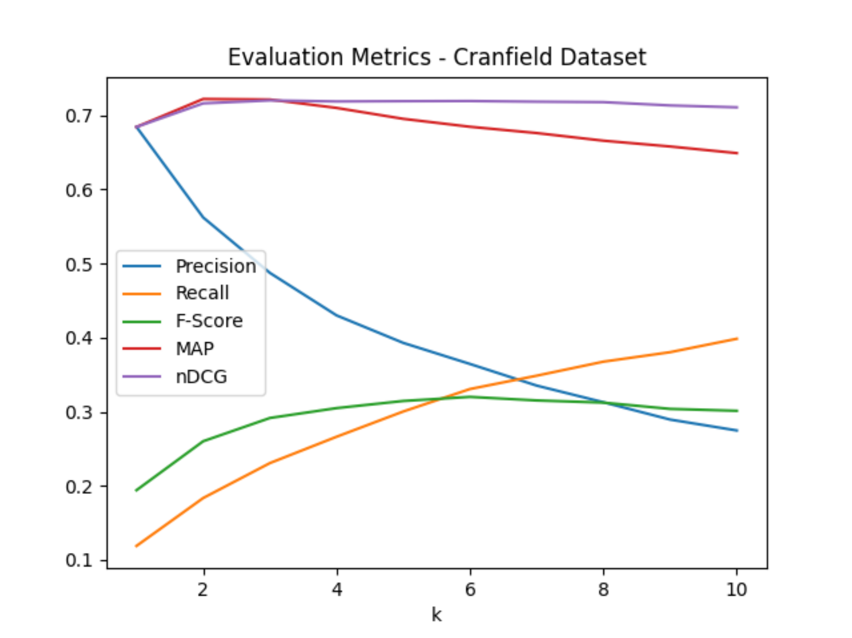


BIGRAM BASED SEARCH

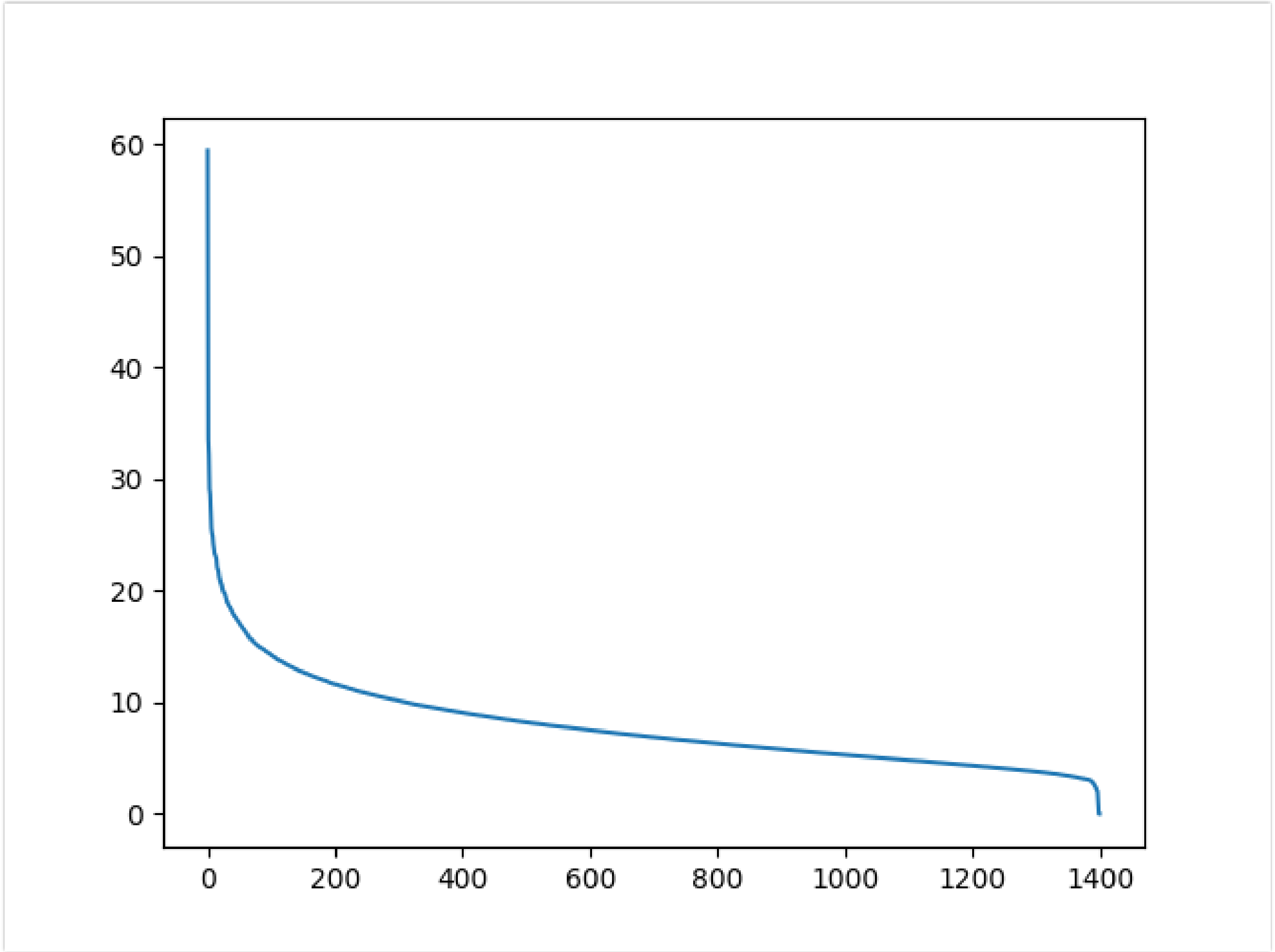
WITHOUT LSA (M=10739)

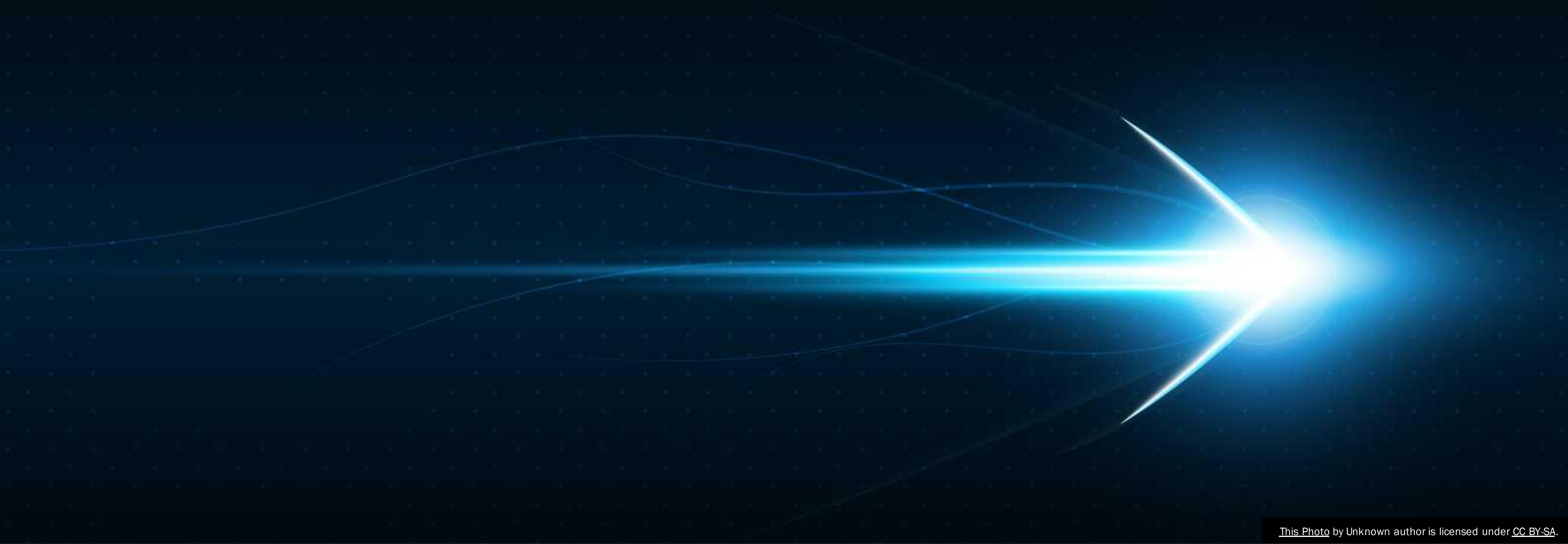


WITH LSA (K=500)



EIGEN VALUES IN LSA over BIGRAM MODEL





EFFICIENCY

QUERY EXECUTION TIME: ESA MODEL

	NO LSA (seconds)	LSA(seconds)
Mean	0.009516	0.008722
Variance	4.03E-08	8.18E-07
Observations	225	225
Hypothesized Mean Difference	0	
df	246	
t Stat	12.85027	
P(T<=t) one-tail	1.47E-29	
t Critical one-tail	1.651071	
P(T<=t) two-tail	2.93E-29	
t Critical two-tail	1.969654	

QUERY EXECUTION TIME: NESA MODEL

	NO LSA (seconds)	LSA(seconds)
Mean	0.012364	0.01187
Variance	1.53E-07	1.25E-06
Observations	225	225
Hypothesized Mean Difference	0	
df	278	
t Stat	6.256535	
P(T<=t) one-tail	7.41E-10	
t Critical one-tail	1.650353	
P(T<=t) two-tail	1.48E-09	
t Critical two-tail	1.968534	

QUERY EXECUTION TIME: BIGRAM MODEL

	NO LSA	LSA
Mean	0.067972	0.007655
Variance	2.87E-06	6.4E-07
Observations	225	225
Hypothesized Mean Difference	0	
df	319	
t Stat	482.5636	
P(T<=t) one-tail	0	
t Critical one-tail	1.649644	
P(T<=t) two-tail	0	
t Critical two-tail	1.967428	

BEST MODEL

EFFICIENCY

Time in sec	bigram	ESA	NESA
Mean	0.007655	0.008722	0.01187
Variance	6.4E-07	8.18E-07	1.25E-06
Observations	225	225	225
Hypothesized Mean Difference		0	0
df		441	406
t Stat		-13.2579	-46.0276
P(T<=t) one-tail		2.66E-34	1.7E-163
t Critical one-tail		1.648316	1.648615
P(T<=t) two-tail		5.32E-34	3.3E-163
t Critical two-tail		1.965358	1.965824

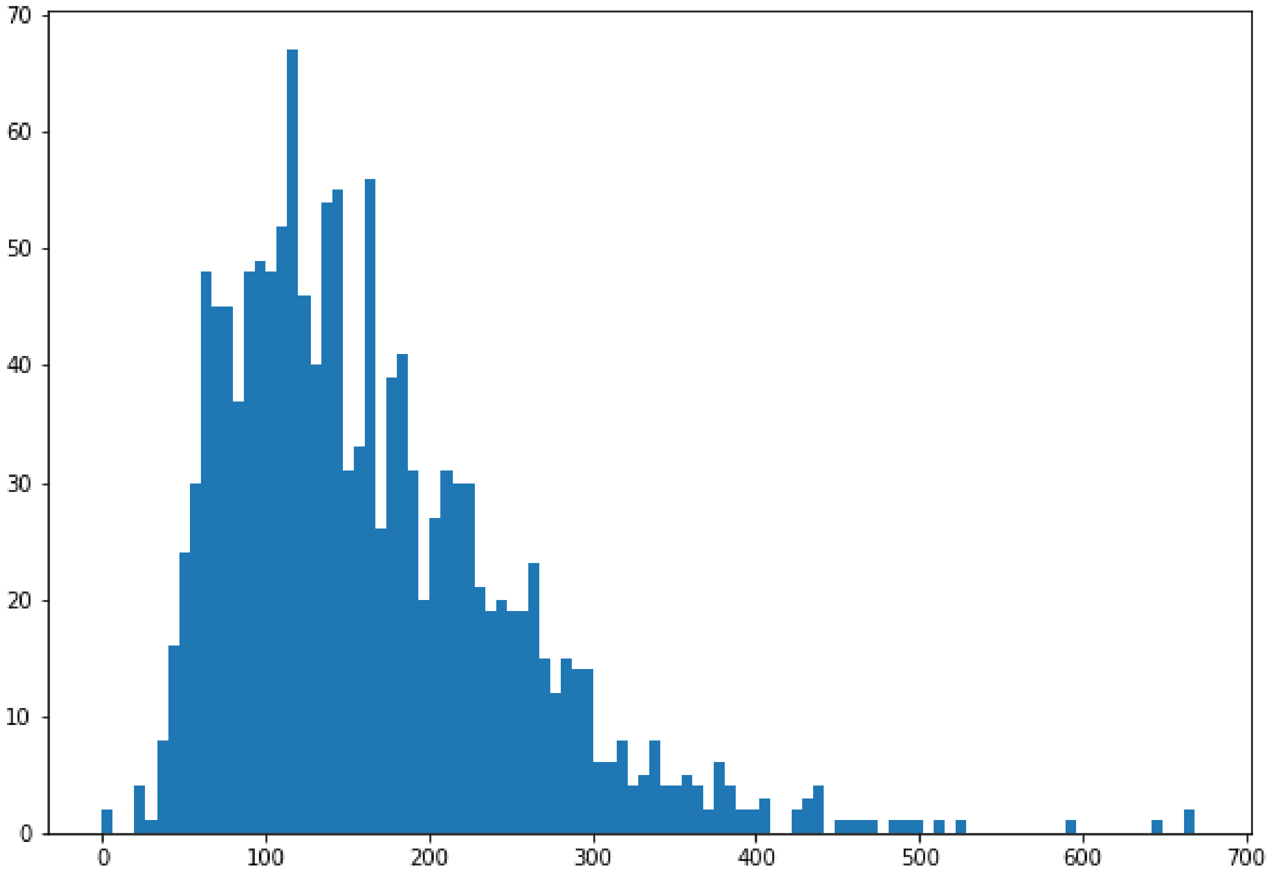
EFFECTIVENESS

nDCG@5	bigram	ESA		ESA	NESA
Mean	0.719252	0.522305	Mean	0.522305	0.528636
Variance	0.125721	0.158199	Variance	0.158199	0.163539
Observations	225	225	Observations	225	225
Hypothesized Mean Difference	0		Hypothesized Mean Difference	0	
df	442		df	448	
t Stat	5.544217		t Stat	-0.1674	
P(T<=t) one-tail	2.54E-08		P(T<=t) one-tail	0.433566	
t Critical one-tail	1.648308		t Critical one-tail	1.648262	
P(T<=t) two-tail	5.09E-08		P(T<=t) two-tail	0.867131	
t Critical two-tail	1.965346		t Critical two-tail	1.965273	

CONCLUSIONS

THE RESULTS OF THE ABOVE EXPERIMENTS CONCLUDE BOTH OUR HYPOTHESES:

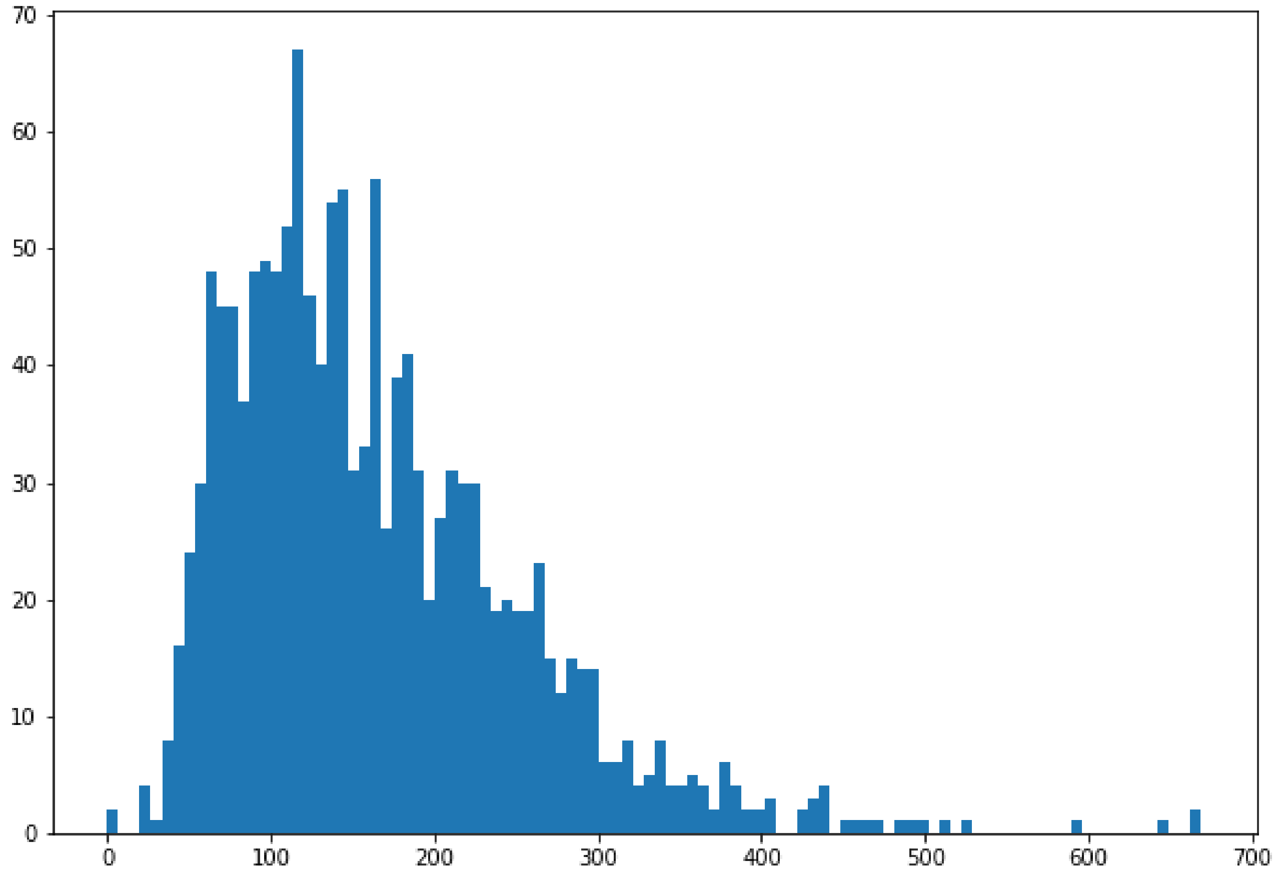
- LATENT SEMANTIC ANALYSIS MAKES MODELS FASTER WHILE EITHER IMPROVING THE EFFECTIVENESS OF THE MODEL OR KEEPING IT THE SAME
- "BIGRAM" VECTOR MODEL WOULD CONSIDER WORD ORDER IN THE CORPUS AND GIVE BETTER RESULTS THAN ESA AND NESA FOR THE CRANSFIELD DATASET



CONCLUSIONS

In other words:

1. Model+LSA is faster in query execution and than A1 on metric E1 on task T without affecting E2 on dataset D under assumption S.



THANK YOU