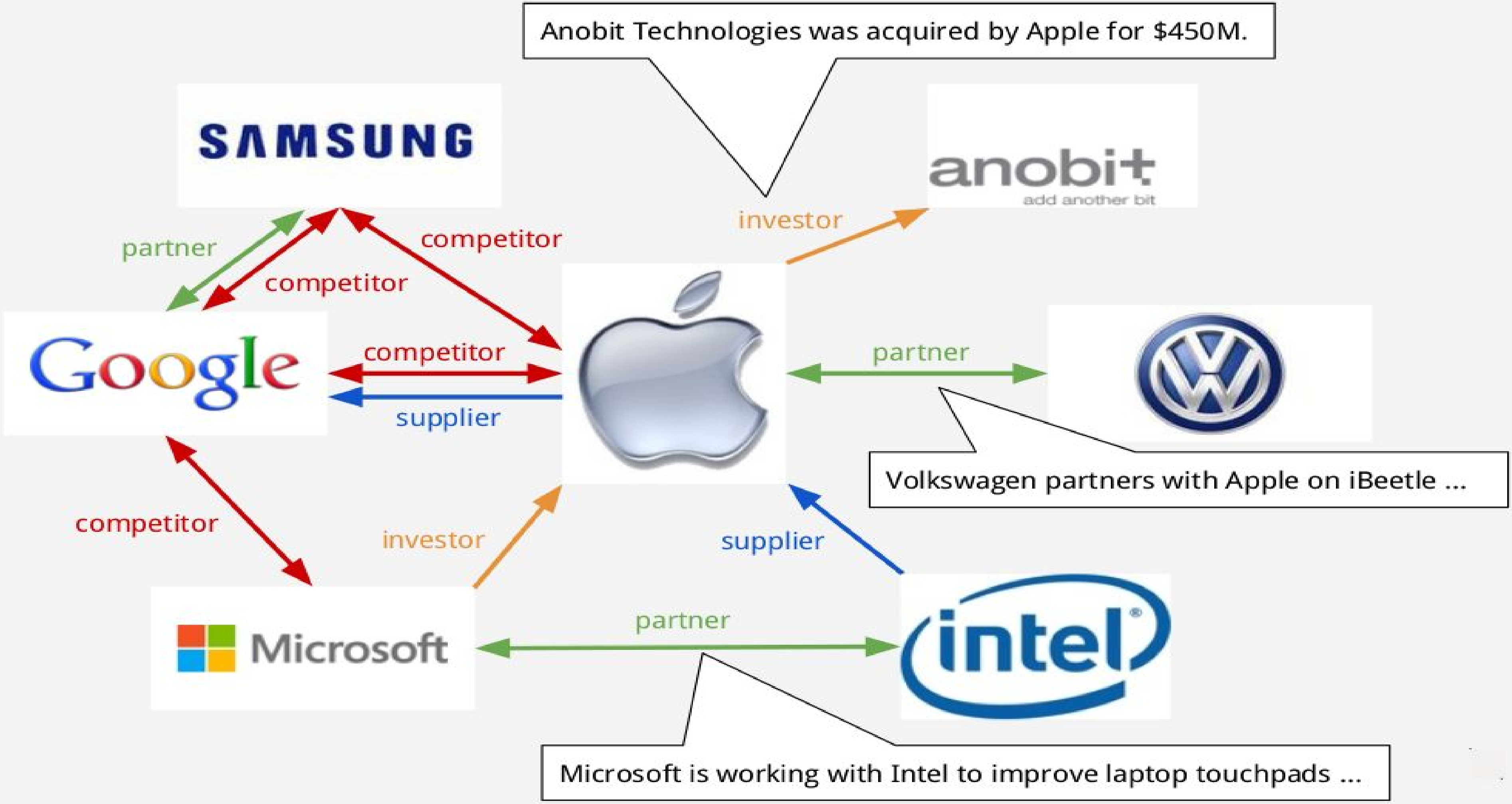
CONCEPT MAP

Presented By -
Sandip Ghoshal (143050024)
Ankith M S (143059007)

CONCEPT MAP

A **concept map** is a way of representing relationships between **concepts**. In a concept map, each concept is linked to another concept and gives visual representation of the hierarchy of the concepts .



Courtesy: http://web.stanford.edu/

PROBLEM STATEMENT

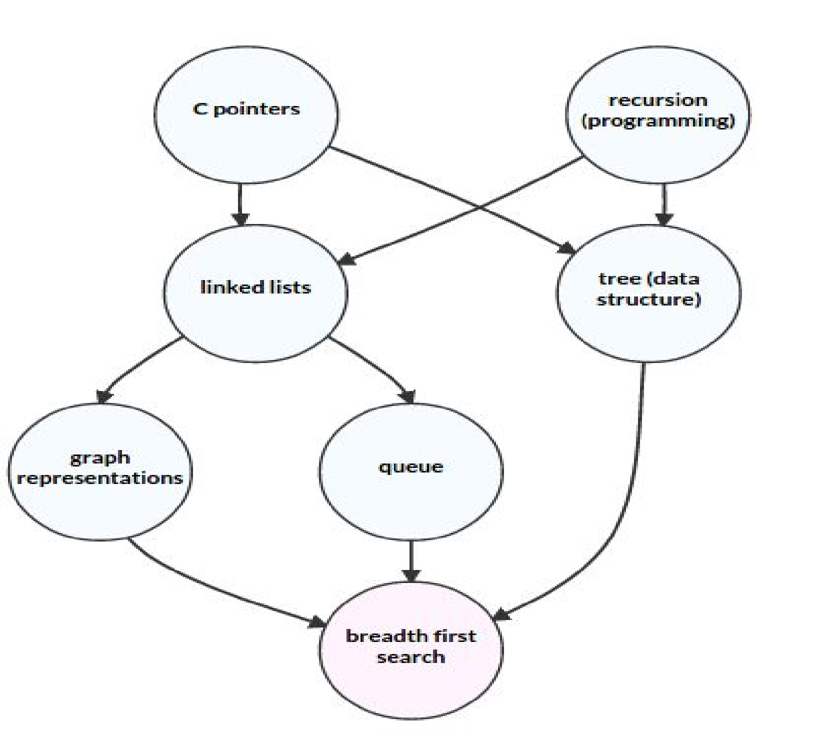
We want to extract the relation among **academic entities** from natural language text (such as text books , educational sites etc.) and create a concept map

Motivation

- Concept map gives the concise representation of the concepts . By grouping facts and concepts .

- Concept map explains the connection between the concepts, helps students to organize and structure their thoughts to understand the domain overview.

CONCEPT MAP - BFS



Approach

- Rule Based
- Supervised Learning
- Distant Supervision

Rule Based Approach

Simple rule for "**part of**" relation :

X is a [**part of | module of**] **Y**

Eg: **Data mining** is a **part of machine learning**

Rule Based - Problems

- Requires hand-building patterns for each relation.
  - hard to write; hard to maintain
  - there are zillions of them
  - domain-dependent

- Low **recall** rate

  Eg : Relation extraction is a **sub-task** of information extraction

Approach

- Rule Based
- Supervised Learning
- Distant Supervision

Supervised Learning

- Need a completely annotated corpus

- Example –
  Sentence - “**HashMaps** are implemented using a data structure known as a **hash table**”
  Relation - **implemented_as**(HashMaps , hash table)

- The corpus is used as the training set.

- Learn features from the corpus.

- Use those features at the time of extraction

Supervised Learning - Problems

- It is **expensive** to produce the training data.

- Very much **domain specific**.

  For example, if we our training domain is **news** , we can not extract relations in **academic** domain.

Approach

- Rule Based
- Supervised Learning
- Distant Supervision

Distant Supervision

- Supervised by a knowledge base.
- Large number of existing relation instances from the KB are used for seeding
- Train on large corpus having sentences that are expressing those relations from the KB
- Extract features from training sentences.
- Use those features at the time of extraction.

Distant Supervision-Better Approach

- No hard coded rules
- No annotated corpus is required
- Supervised by an existing knowledge base
- No limitation in training examples

Concept Mapping-Distant Supervision Approach

Step-1 – Choose a Knowledge Base.
Step-2 – Select relations to train the model.
Step-3 – Select existing relation instances from the KB.
Step-4 – Find all the sentences in the training corpus that contains the entities from the KB.
Step-5 – Model to select sentences that expresses the relation among entities.
Step-6 – Extract Features from training sentences.

Knowledge Base

- A repository of structured information about entities.
- Contains set of entities, relations among entities and type hierarchies.

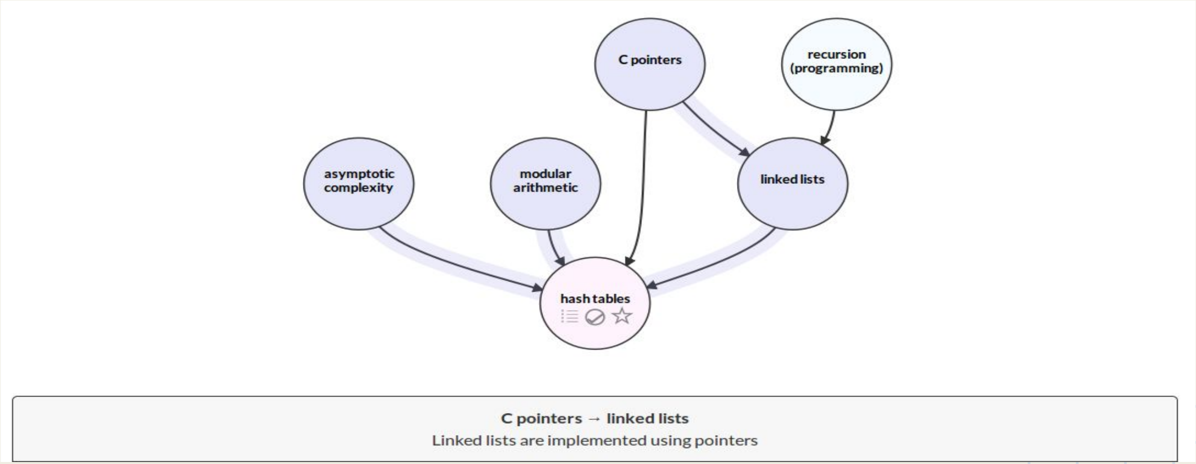
Concept Map- Knowledge Base

Metacademy

Efficient knowledge base for academic domain

Why Metaacademy

- Lack of knowledge base for educational domain .
- Metaacademy uses trivial sentences(simple dependency path) to express the relation between entities



Concept Map- Knowledge Base

Our system requires the Knowledge Base to be expressed in the form of **triplets** < ‘concept1’ , ‘concept2’ , ‘relation name’ >

Eg: < ‘HashMaps’ , ‘Hash Table’ , ‘implemented as’ >

Concept Map- Knowledge Base

- Number of distinct examples extracted - **1189**
- No. of concepts - **472**

Concept Map- Knowledge Base

- Clustering relations
  - Manual

Relation Name	Keywords
algo_for	algorithm for, algorithm to, algorithms for
approx_to	approximation to
computed_using	to compute, computed using, computes, for computing the, way of computing

https://drive.google.com/drive/u/0/folders/0B36u0U_YDCzQY3JEMkFZTHgzRTA

Concept Map- Knowledge Base

- Clustering relations - Another approach
    - **Word2vec**
        - Word2Vec is a google tool which gives similarity between two words.
        - We trained word2vec with different values of vector length using our book - wikipedia corpus as training data and clustered relations having similarity score greater than 0.5

Concept Map- Knowledge Base

- Clustered relations
    - Word2vec

Relation Name	Keywords
algo_for	technique,method,algorithm
involve	involves ,requires,needed
typically	often,typically

Concept Mapping-Distant Supervision Approach

Step-1 – Choose a Knowledge Base.
Step-2 – Select relations to train the model.
Step-3 – Select existing relation instances from the KB.
Step-4 – Find all the sentences in the training corpus that contains the entities from the KB.
Step-5 – Model to select sentences that expresses the relation among entities.
Step-6 – Extract Features from training sentences.

Concept Mapping - Relation selection

- Distant supervision requires large number of sentences for each relation to train the model

- We choose the relations with sufficient number of instances

Relations from KB

Relation Name	Keywords	# Examples
algo_for	algorithm for, algorithm to, algorithms for	22
approx_to	approximation to	8
computed_using	to compute, computed, computes, for computing	21
def_in_terms_of	define, defined, defined as, defines, defining, definition, definition of	55
example_of	example for, example of, examples of, instance of, is an instance, case of	25
gen_of	generalization of, generalizes	14
implemented_as	implementation of, implemented as, implemented using, are often implemented, often implemented	19
part_of	part of	15
prop_of	is a property, property of	9
rep_as	represent, are represented, be represented, represented, represented as, represented in	15
spl_case_of	a special case, is a special case, special case, special case of	4

Concept Mapping-Distant Supervision Approach

Step-1 – Choose a Knowledge Base.
Step-2 – Select relations to train the model.
Step-3 – Select existing relation instances from the KB.
Step-4 – Find all the sentences in the training corpus that contains the entities from the KB.
Step-5 – Model to select sentences that expresses the relation among entities.
Step-6 – Extract Features from training sentences.

Concept Mapping-Training Corpus

Corpus :

- Text Book
  - Bishop-Pattern Recognition and Machine Learning
  - Algorithms in C++

- Wikipedia
  - Extracted pages from wikipedia categories, starting with Category:Computer_science and upto 6 level of category hierarch

Training data generation - PDF to Text

Text book corpus :

- Need to convert pdf documents to plain text format .
  - PDF to Txt tools
    - PdfMiner
    - General Architecture for Text Engineering
    - Zamzar
    - PdfBox

Since we used technical books,only pdfbox was able to identify text, equations and code snippets as different .

Training data generation - PDF to Text

- **Pdf to text** conversion is **noisy** :
  - Special characters handling
  - Hard to **sentencify** the text
    - Page headers and footers appended in sentences
    - Difficult to distinguish code segment
    - Table data and image captions introduce noise too
- Used NLTK tool for sentencification.

Concept Mapping-Training

Used NLTK tool for sentencification

Eg:

"*With other branches of mathematics it has grown beyond the circumstances of its birth.*"

NltK output :

["*With other branches of mathematics it has grown beyond* "]

["*the circumstances of its birth.*"]

We parsed the output again to stitch the two elements into a single one.

["*With other branches of mathematics it has grown beyond the circumstances of its birth.*"]

Processing the KB

- Clean up and lemmatize the concept names
- **Example -**
  - “classes (programming)” → [“classes “]
  - “Zorn’s Lemma” → [“Zorn’s Lemma”, “Zorns Lemma”]
  - “alpha-beta pruning” → [“alpha-beta pruning”, “alpha beta pruning”]
  - “Data structure: Stack” → [“Stack”]
  - “Stacks” → [“Stack”, “Stacks”]

Processing the KB

- Rewrite the KB with modified concept names with all possible combinations

- **Example** -
  - Original KB - <”dot products”, “convex sets”, “example_of”>
  - Modified KB -
    - <”dot product”, “convex set”, “example_of”>
    - <”dot products”, “convex set”, “example_of”>
    - <”dot product”, “convex sets”, “example_of”>
    - <”dot products”, “convex sets”, “example_of”>

Processing the KB

Comparison -

	Old KB	Modified KB
No. of Triplets	207	511

Processing the KB

- Find distinct concept names from the KB
- Generate an id for each concept
- Write the concepts with ids in a different file called **concept_id.txt**
- **Example** -
  - KB -
    - <hidden Markov models particle filter algo_for>
    - <hidden Markov models forward-backward algorithm algo_for>
  - concept_id.txt -
    - 1 hidden Markov models
    - 2 particle filter
    - 3 forward-backward algorithm

Processing the KB

- Generate a new KB with concept ids rather than concept names.

- **Example** -
  - **Original KB** -
    - <hidden Markov models   particle filter   algo_for>
    - <hidden Markov models   forward-backward algorithm   algo_for>
  - **concept_id.txt** -
    - 1   hidden Markov models
    - 2   particle filter
    - 3   forward-backward algorithm
  - New **KB_Id.txt** -
    - <1   2   algo_for>
    - <1   3   algo_for>

Processing the KB

- Read the corpus
- Find out sentences where any two entities from the corpus are present
- From books we found 6957 such sentences
- Assign each such sentence an unique id.

Processing the KB

- Select sentences that contains entities that are
- Find out sentences where two entities participate in a relation are present
- Assign each such sentence an unique id.

Processing the KB

- Feed for the “**NA**” relation

- As we want to train our model with a specific set of relations it is important to train the model with the “NA” relation, for any relation outside our choice

- We choose sentences that contain unrelated entities as a feed for the “NA” training

Processing the KB

- Generate the **Match File**
- We need to feed this file to Multi-R
- Match file uses the entity and sentence ids we assigned earlier.
- Example Match File entry

Sentence - *“**Stack** can be implemented using **linked lists**”*

<ent1_id>	<ent1_start>	<ent1_end>	<ent1>	<ent2_id>	<ent2_start>	<ent2_end>	<ent2>	<sentence_id>	<relation>
1	1	5	Stack	2	31	42	linked lists	100	implemented_as

Named Entity Recognition

- Named Entity Recognition is an important step to find the entity mentions in the corpus

- (NER) is a task to locate and classify elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

- **Example**
  - **Ramanujan_PERSON was born in India_PLACE.**

Named Entity Recognition

- It is hard to perform the task of NER in to tag educational entity mentions
- No such well known tagger exists
- We take a step ahead and try **Named Entity Linking**

Named Entity Linking (NEL)

- **Named Entity Linking** (NEL) is the task of identifying entities in natural language text and map them to an existing KB.

- We used the **Yago** database to map the **concept name** (entity mentions) to their wiki category.

- **Example** -
    - **avl tree** - <wordnet_data_structure_105728493>

- Problems - multiple categories associated to one concept
    - **Counter-Strike** - <wikicategory_Electronic_sports_games>
      <wikicategory_Multiplayer_online_games>

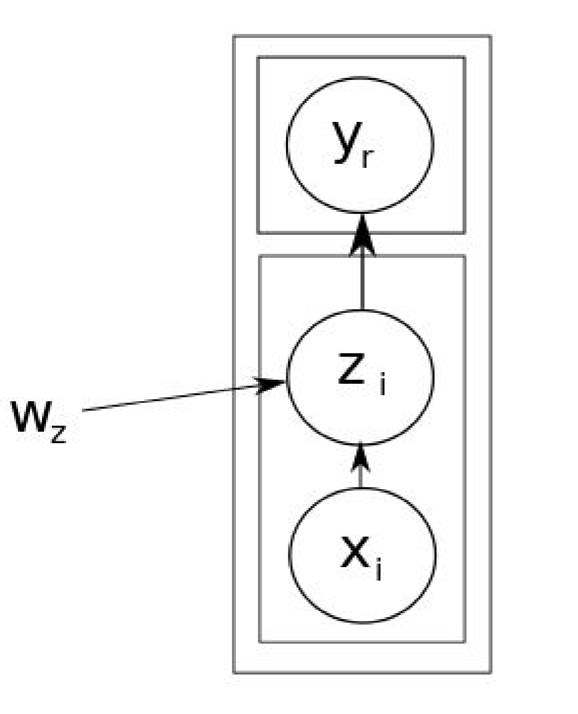
Concept Mapping-Distant Supervision Approach

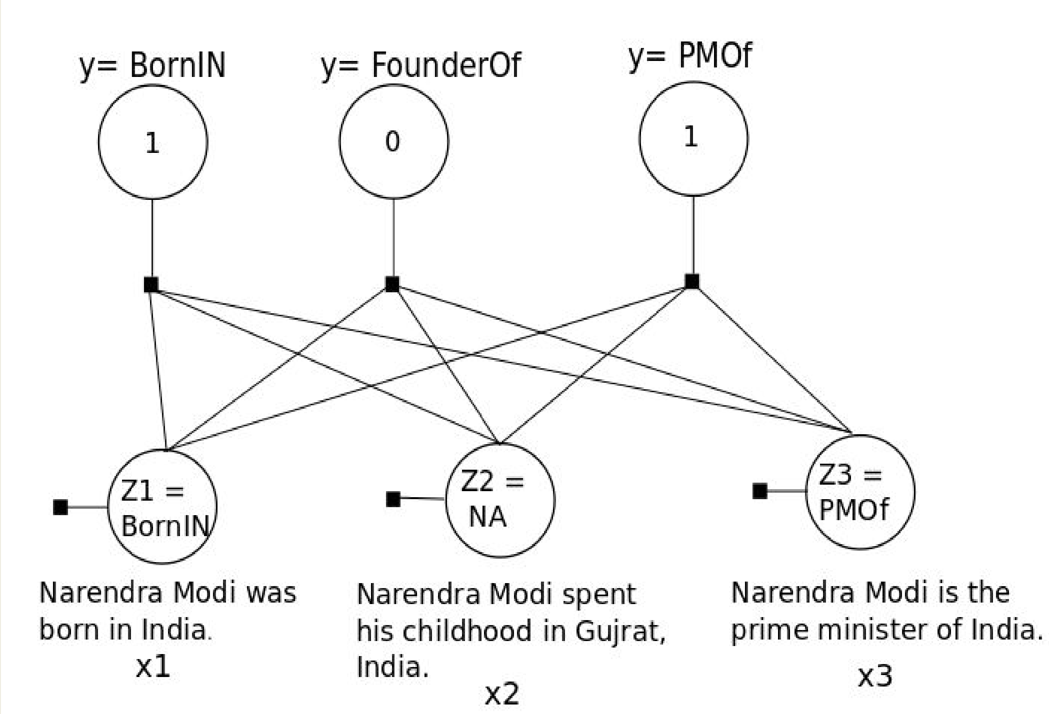
Step-1 – Choose a Knowledge Base.
Step-2 – Select relations to train the model.
Step-3 – Select existing relation instances from the KB.
Step-4 – Find all the sentences in the training corpus that contains the entities from the KB.
Step-5 – Model to select sentences that expresses the relation among entities.
Step-6 – Extract Features from training sentences.

Multi-R

- **Multi-R -** This model was proposed by Hoffman et. al. (2011)
- Multi Instance Multi Label Learning
- A graphical approach
- The **assumption** of the model is if two entities **e1** and **e2** are related through relation **r** in the KB.Then the entities **e1** and **e2** appears together in at least one sentence in the corpus, that expresses the relation.

Multi-R - A Graphical Model





Concept Mapping-Distant Supervision Approach

Step-1 – Choose a Knowledge Base.
Step-2 – Select relations to train the model.
Step-3 – Select existing relation instances from the KB.
Step-4 – Find all the sentences in the training corpus that contains the entities from the KB.
Step-5 – Model to select sentences that expresses the relation among entities.
Step-6 – Extract Features from training sentences.

Features

- Features should be selected such a way that it should describes how two entities are related in a sentence, using either syntactic or semantic information.

Types of Features

- Mintz et. al. (2009)
  - Lexical Features
  - Semantic Features

Lexical Features

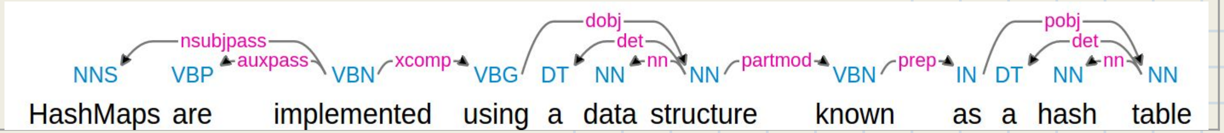
- **Lexical Features - sequence of words between the two entities**
  - The word sequence between the two entities
  - The part-of-speech tags of the words
  - A flag indicating which entity came first in the sentence
  - A window of k words to the left of Entity 1 and their part-of-speech tags
  - A window of k words to the right of Entity 2 and their part-of-speech tags

Semantic Features

- **Semantic Features -**
  - **To find the Semantic feature we use the “Dependency Parse Tree” of the sentence.**

MultiR Features

Eg:“HashMaps are implemented using a data structure known as a hash table ”

Entity strings	HashMaps,Hash table
Bag of words in entities	HashMaps,Hash Table,Hash,Table
bigram feature between Entity Pairs	{implemented using, using data,data structure}
Syntactic structure	HashMaps|NNP are|VBP implemented|VBN using|VBG a|DT data|NN structure|NN known|VBN as|IN a|DT hash|NN table|NN
Dependency path	

Using Word Embeddings as features of MultiR

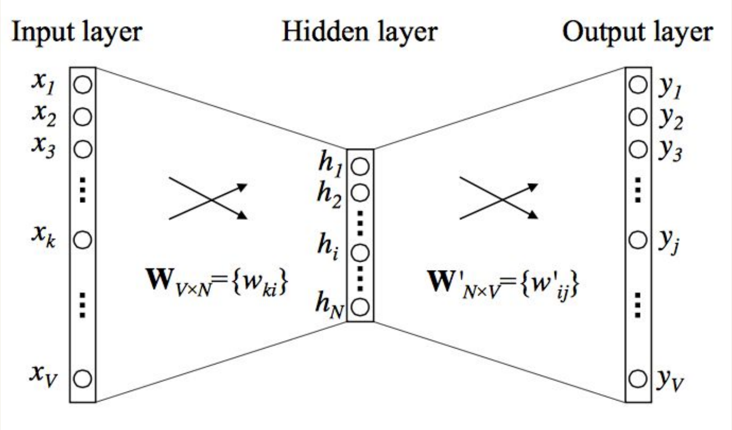
Introduction to Word2Vec

- Word2Vec is a tool where words or phrases from the vocabulary are mapped to vectors of real numbers in a low dimensional space, relative to the vocabulary size .

- Vectors of two similar words tends to be the same , so cosine between two similar words tends 1 .

Word2Vec Training

- Input : Text Corpus
- Output : Word Vectors
- Two training models : Cbow and skip-gram



Word2Vec in MultiR features

Convert the phrase between entities to phrase vector

Eg : if there are two sentences in our training corpus as below

- dijkstra **[is an algorithm for]** shortest path
- dijkstra **[is a solution for]** shortest path

The similarity score between “is an algorithm for” and “is a solution for” is 0.634 , so the above two sentence try to express the same relation .

Word2Vec in MultiR features

- Vector of an entities

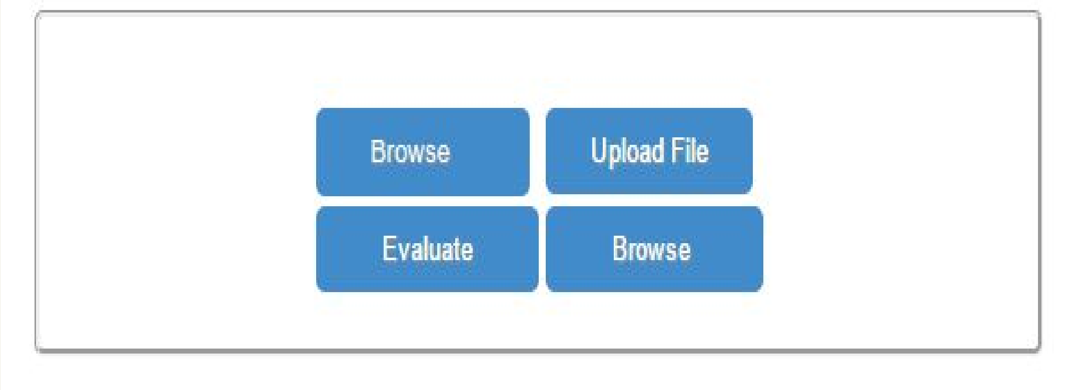
  Eg: Entities such as Bayes, bayesian,classifier,probabilistic,frequentist   all these entities belong to same domain and these vectors are similar to each other .

Evaluation

Evaluator UI

Evaluation - Evaluator UI

- A **tool** created that can be **reused**.
- A link to upload the result file from the Multi-R or other system.
- The format of the result file should be
  <Test Sentence> <Relation(entity1, entity2)>

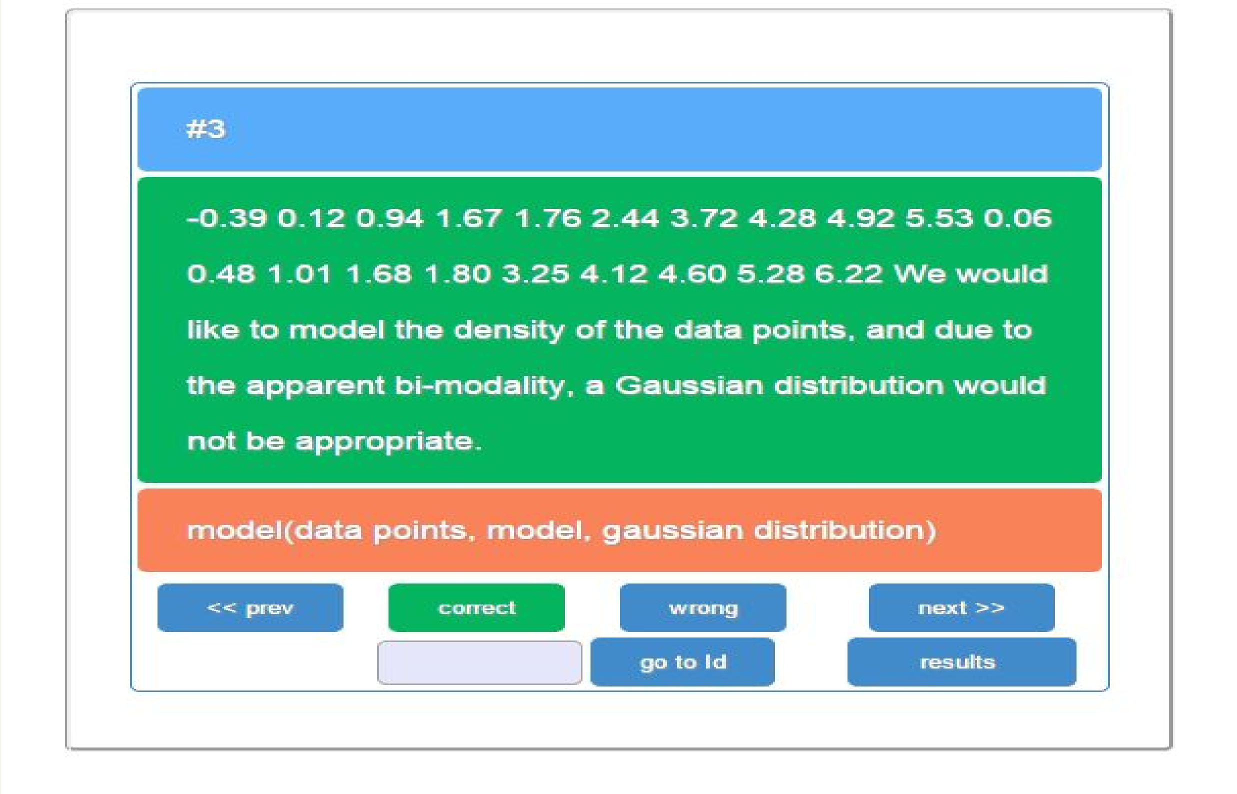


Evaluation - Evaluator UI

- Once the user upload the file to verify, he/she can go the evaluate link.

- In the evaluate screen the sentences will be shown to the user one at a time.

Evaluation - Evaluator UI



Evaluation - Evaluator UI

- Results - Consolidated

Correct	Wrong	Not evaluated
10 (43%)	13 (56%)	177

show all

Evaluation - Evaluator UI

- Results - Individual

#11	1006 CHAPTER 32 SECURITY IN THE INTERNET: IPSec, SSUTLS, PGP, VPN, AND FIREWALLS Private Networks An organization that needs privacy when routing information inside the organization can use a private network as discussed previously.	use(pgp, use, routing information)
#12	1006 CHAPTER 32 SECURITY IN THE INTERNET: IPSec, SSUTLS, PGP, VPN, AND FIREWALLS Private Networks An organization that needs privacy when routing information inside the organization can use a private network as discussed previously.	use(privacy, use, routing information)
#13	1006 CHAPTER 32 SECURITY IN THE INTERNET: IPSec, SSUTLS, PGP, VPN, AND FIREWALLS Private Networks An organization that needs privacy when routing information inside the organization can use a private network as discussed previously.	use(private network, use, routing information)
#14	1006 CHAPTER 32 SECURITY IN THE INTERNET: IPSec, SSUTLS, PGP, VPN, AND FIREWALLS Private Networks An organization that needs privacy when routing information inside the organization can use a private network as discussed previously.	use(routing information, use, private network)
#15	1006 CHAPTER 32 SECURITY IN THE INTERNET: IPSec, SSUTLS, PGP, VPN, AND FIREWALLS Private Networks An organization that needs privacy when routing information inside the organization can use a private network as discussed previously.	use(vpn, use, routing information)
#16	100Base-FX The two-wire fiber implementation of Fast Ethernet.	implementation(fiber, implementation, fast ethernet)
#17	100Base-FX The two-wire fiber implementation of Fast Ethernet.	implementation(two-wire, implementation, fiber)
#18	: (10.10) When specialized to the case in which 0 and 1 are path homotopic, this implies the theorem, because 0 and 1 are constant maps in that case.	example(homotopic, case, theorem)
#19	10.11 Describe what happens on the second level of partitioning (when the left subfile is partitioned and when the right subfile is partitioned) when we use binary quicksort to sort a random permutation of the nonnegative integers less than 171.	use(partitioned, use, quicksort)
#20	10.11 Describe what happens on the second level of partitioning (when the left subfile is partitioned and when the right subfile is partitioned) when we use binary quicksort to sort a random permutation of the nonnegative integers less than 171.	use(partitioning, use, quicksort)

Evaluator UI

Information Retrieval

Information retrieval is the process of fetching documents within the large collection of documents that satisfies the give the given queries

Information Retreival

In top-k system , queries are evaluated using two major families of algorithms

- Term at a time(TAAT)
- Document at a time (DAAT)

DAAT

Document At A Time

Introduction

Document-at-a-time (DAAT) strategies evaluate the contributions of every query term with respect to a single document before moving to the next document.

Author’s approach

Efficient Query Evaluation using a Two-Level Retrieval Process

WHY DAAT

- DAAT implementations require a smaller run-time memory
- DAAT exploit I/O parallelism more effectively by traversing postings lists on different disk drives simultaneously

Scoring

$$\text{Score}(d, q) = \sum_{t \in q \cap d} \alpha_t w(t, d)$$

α_t is a function of the number of occurrences of t in the query, multiplied by the inverse document frequency of t in the index
w(t, d) is a function of the term frequency (tf) of t in d, divided by the document length |d|.

Scoring

$$UB_t \geq \alpha_t \max(w(t, d_1), w(t, d_2), \ldots).$$

$$UB(d, q) = \sum_{t \in q \cap d} UB_t \geq Score(d, q).$$

Preliminary scoring

$$\mathbf{WAND}(X_1, UB_1, X_2, UB_2, \ldots, X_k, UB_k, \theta)$$

where Xi is an indicator variable for the presence of query term i in document d
The threshold θ is set dynamically

In practise **$\theta = F \cdot m$**

DAAT-Algorithm

```
Function init(queryTerms)
    terms ← queryTerms
    curDoc ← 0
    for each t ∈ terms
        posting[t] ← t.iterator.next(0)
```

Sets the current document to be considered (curDoc) to zero and for each query term, t, it initializes its current posting posting[t] to be the first posting element in its posting list

DAAT-Algorithm

After init , next method is called repeatedly which takes threshold θ and returns the next document whose approximate score is larger than θ

```
Function next(θ)
    repeat
        /* Sort the terms in non decreasing order of
         DID */
        sort(terms, posting)
        /* Find pivot term - the first one with accumulated
         UB ≥ θ */
        pTerm ← findPivotTerm(terms, θ)
        if (pTerm = null) return (NoMoreDocs)
        pivot ← posting[pTerm].DID
        if (pivot = lastID) return (NoMoreDocs)
        if (pivot ≤ curDoc)
            /* pivot has already been considered, advance
             one of the preceding terms */
            aterm ← pickTerm(terms[0..pTerm])
            posting[aterm] ← aterm.iterator.next(curDoc+1)
        else /* pivot > curDoc */
            if (posting[0].DID = pivot)
                /* Success, all terms preceding pTerm belong
                 to the pivot */
                curDoc ← pivot
                return (curDoc, posting)
            else
                /* not enough mass yet on pivot, advance
                 one of the preceding terms */
                aterm ← pickTerm(terms[0..pTerm])
                posting[aterm] ← aterm.iterator.next(pivot)
    end repeat
```

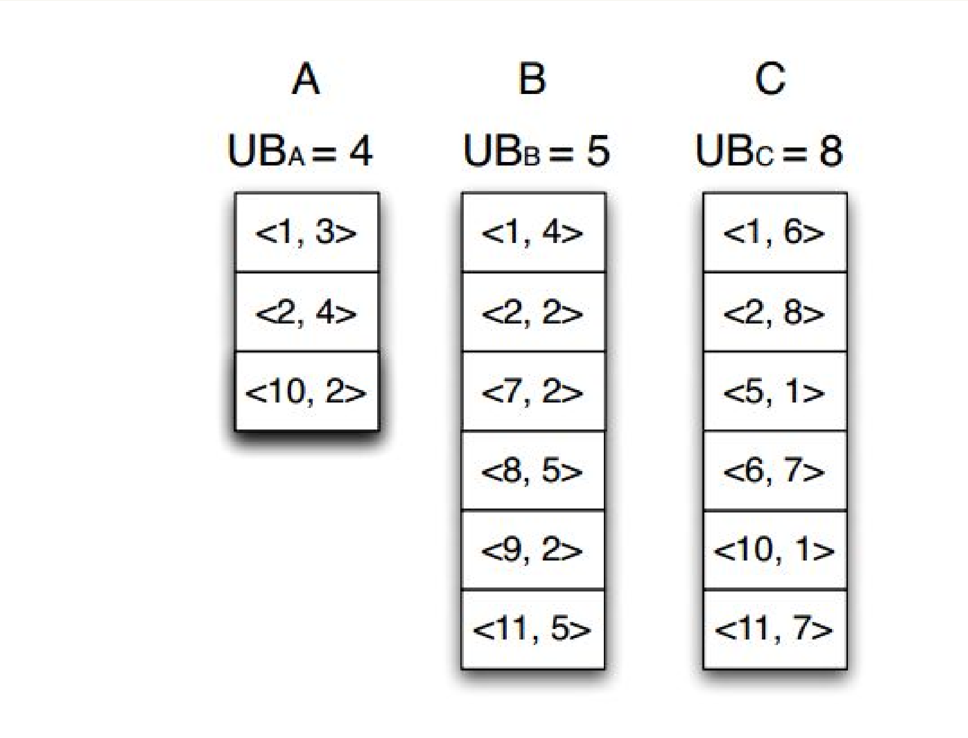
Example

Let us consider the following query

Q={A,B,C}

k=2 , i.e need to fetch top 2 documents satisfying the query.

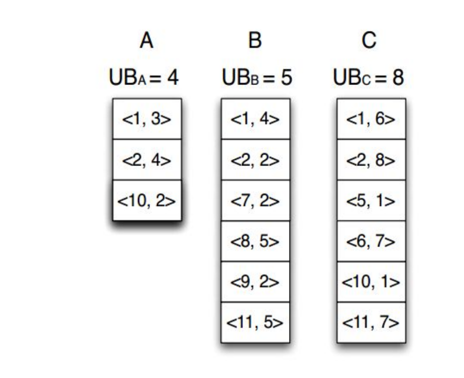
Example - Posting list



Example

After processing doc 1 and 2

Heap	
docid	$score(d, Q)$
1	13 (θ)
2	14



Next step is to sort the cursor by their docid i.e after processing doc 1 and 2 , cursor of query terms A,B,C points to 10,7.5 respectively . So after sort we have

	C	B	A
p	1	2	3
docid	5	7	10

Example

Select pivot document

For $p = 1$, we have:

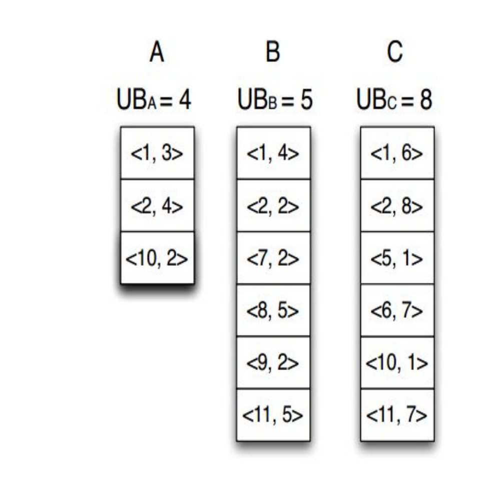
$$UB_C = 8 < \theta = 13$$

For $p = 2$ we have:

$$UB_C + UB_B = 8 + 5 = \theta = 13$$

For $p = 3$ we have:

$$UB_C + UB_B + UB_A = 8 + 5 + 4 > \theta = 13$$



So docid 10 has been selected as pivot

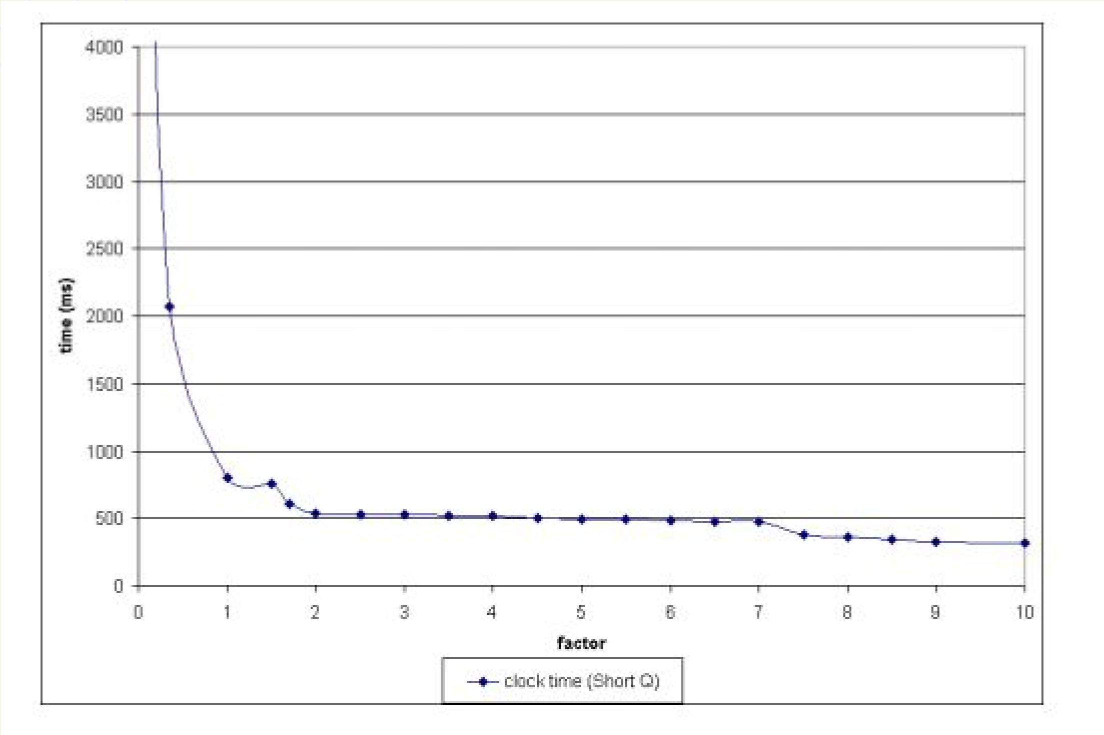
Example

- This means that the minimum docid that can potentially be in the top-k results is document 10.
- Next step is to move one of the query terms to 10 in order to continue processing.
- When compared to the naive DAAT algorithm, it is clear that WAND may reduce both the index access and the scoring costs. In this simple example docids 6, 8 and 9 are completely skipped in postings list for terms B and C and are not scored.

WAND Threshold

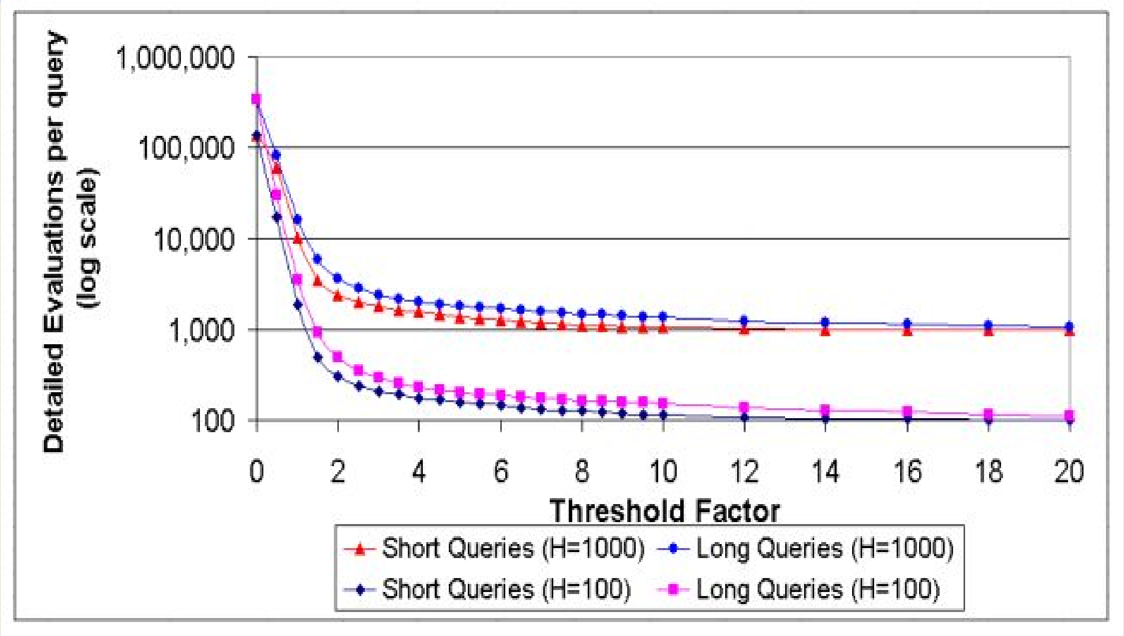
- Initial threshold
  - ‘0’ or ‘sum of all term upper bounds’ or ‘something else’?
- To handle mandatory terms
  - set to some huge value, H

EXPERIMENTAL RESULTS



The average query time for short queries as a function of the threshold factor (heap size H=1000).

EXPERIMENTAL RESULTS



The number of full evaluations as a function of the threshold factor. (H = the heap size.)

EXPERIMENTAL RESULTS

Relative difference

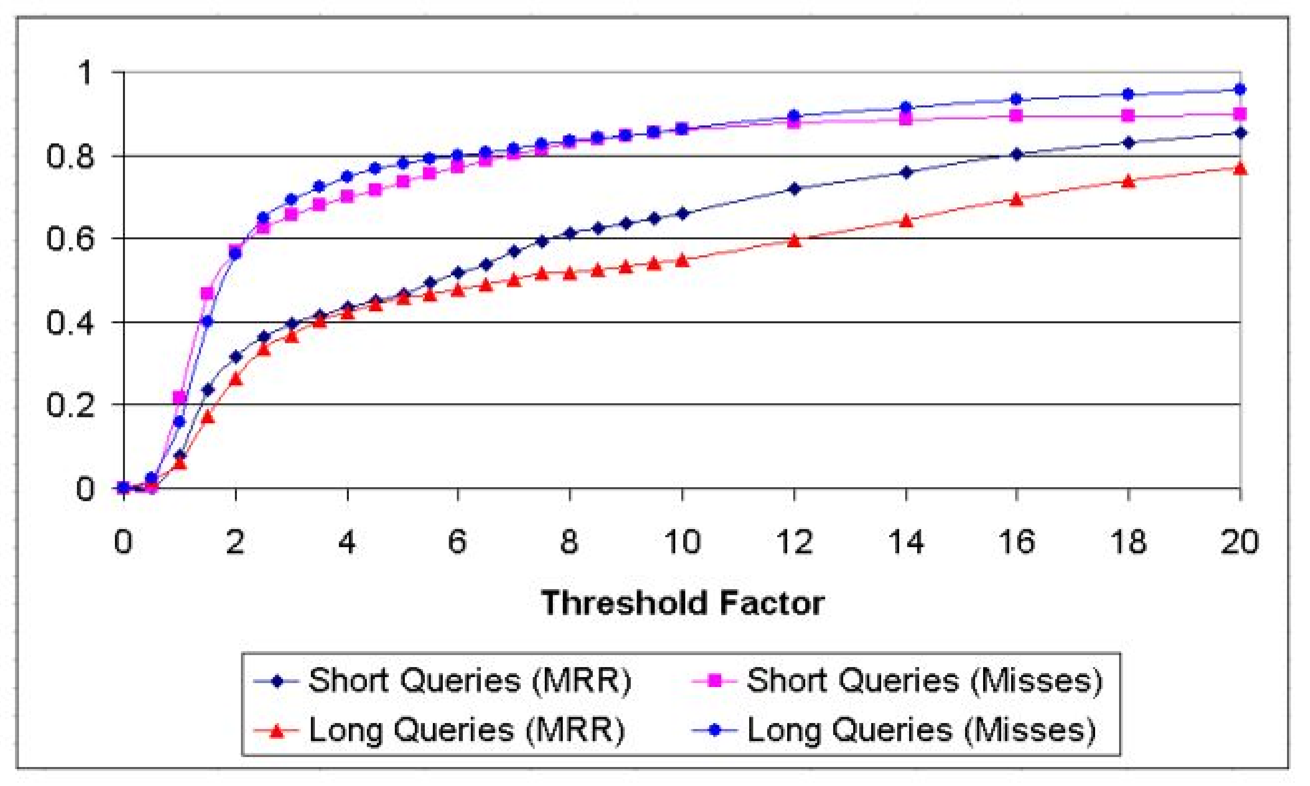
$$\frac{|B \setminus P|}{|B|} = \frac{k-j}{k}$$

MRR (Mean reciprocal rank)

Any document that is in the basic set, B, in position i in the order, but is not a member of the pruned set, P, contributes 1/i to the MRR distance

$$MRR(B,P) = \frac{\sum_{i=1, d_i \in B-P}^{k} 1/i}{\sum_{i=1}^{k} 1/i}$$

EXPERIMENTAL RESULTS



Relative difference (misses) and MRR distance as a function of the threshold factor.

M-WAND

Memory resident WAND

M-WAND

Between mWAND and the original algorithm

After a pivot term p is selected,move all terms between 1 and p beyond the pivot document

M-Wand and WAND

SI	SQ	LQ
Pivot selections (WAND)	2,843.44	17,636.18
Pivot selections (mWAND)	2,840.13	12,798.87
Skipped postings (WAND)	532.56	28,581.22
Skipped postings (mWAND)	531.20	27,214.16
Latency (WAND)	206.0	5,519.0
Latency (mWAND)	200.0	2,104.6

LI	SQ	LQ
Pivot selections (WAND)	28,007.55	282,356.02
Pivot selections (mWAND)	27,814.06	275,164.82
Skipped postings (WAND)	48,089.58	82,511.85
Skipped postings (mWAND)	47,985.65	66,997.41
Latency (WAND)	1896.6	14,082.6
Latency (mWAND)	1867.0	7,556.3

References

Efficient Query Evaluation using a Two-Level Retrieval Process - Andrie Z Border

Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations - Raphael Hoffmann et. al.

Distant supervision for relation extraction without labeled data - mintz et. al.

A Systematic Exploration of the Feature Space for Relation Extraction - Jing Jang

word2vec Explained: Deriving Mikolov et al.’s Negative-Sampling Word-Embedding Method - Mikolov

Thanks !!!