Job Recommendation System Using Deep Learning

Athanasia Farmaki
Academic Supervisor: Dr. Ion Androutsopoulos
Company Supervisor: Iason Andriopoulos
Company : WORKABLE

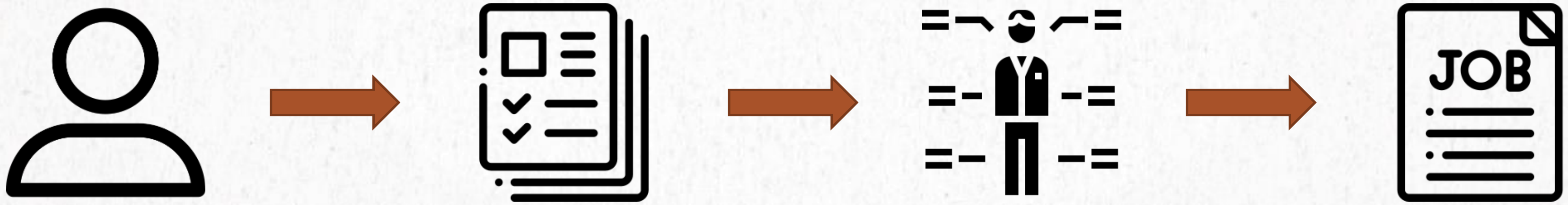
Context

Workable is a recruiting software which connects companies with candidates searching for jobs.

- Provides multiple hiring pipelines, organized candidate profiles, structured interviews and a full reporting suite gives hiring teams the information they need to make the right choice
- Mostly Small and Medium sized enterprises who lack HR department.
- Software provides the functionality of proposing candidates to companies.
- Not the opposite.

Project Goal

- Recommend to candidate the most appropriate job.
- CV not available due to privacy issues
- Information about candidate: previous job applications



AD-HOC RETRIEVAL PROBLEM

- Find documents of a structured, semi-structured or unstructured nature that is relevant to some information given, from within large collection of documents.

- Solution: Search Engine(ElasticSearch)
  - Very fast using inverted index
  - BM25 score

- Inverted Index
  Dictionary of vocabulary words with list of documents as keys.

- BM25 score

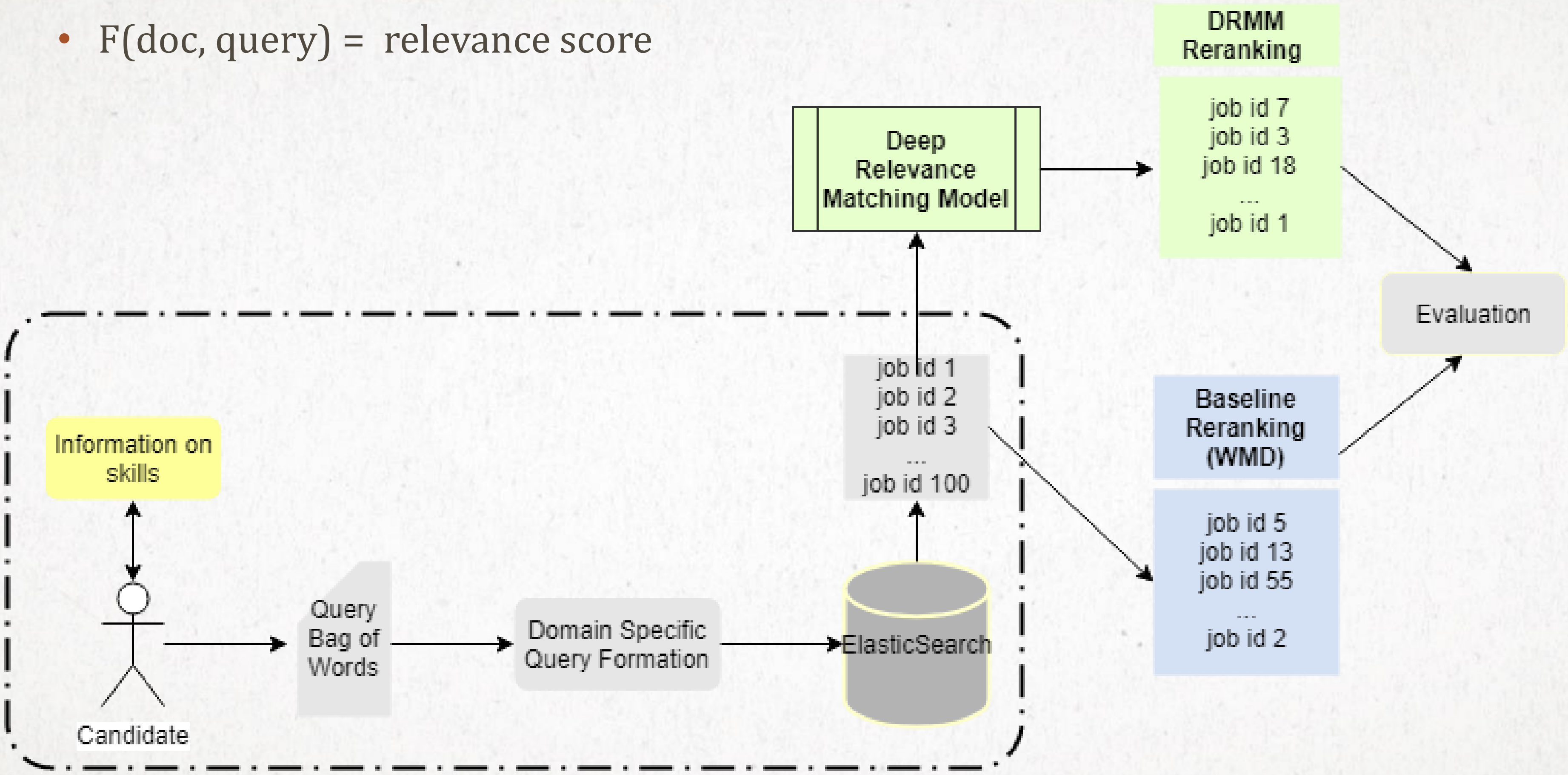
  Takes into consideration:
  - Term Frequency
  - IDF: How rare that term is
  - The size of the document
  - Produces Relevance score

- More sophisticated ways?

Deep Learning Solution

Deep Relevance Matching Model(DRMM)

- F(doc, query) = relevance score



Presentation Outline

- Dataset Presentation
- DRMM explanation
- Word Mover's Distance explanation
- Evaluation Measures
- Data Preprocessing for DRMM
- Experiments - Results

Dataset

- 7.000.000 Candidates
- 400.000 Jobs on database

Candidate data: List of past Job Applications with success score.

Job data: Bag of words per field title, function, description, requirement summary, keywords

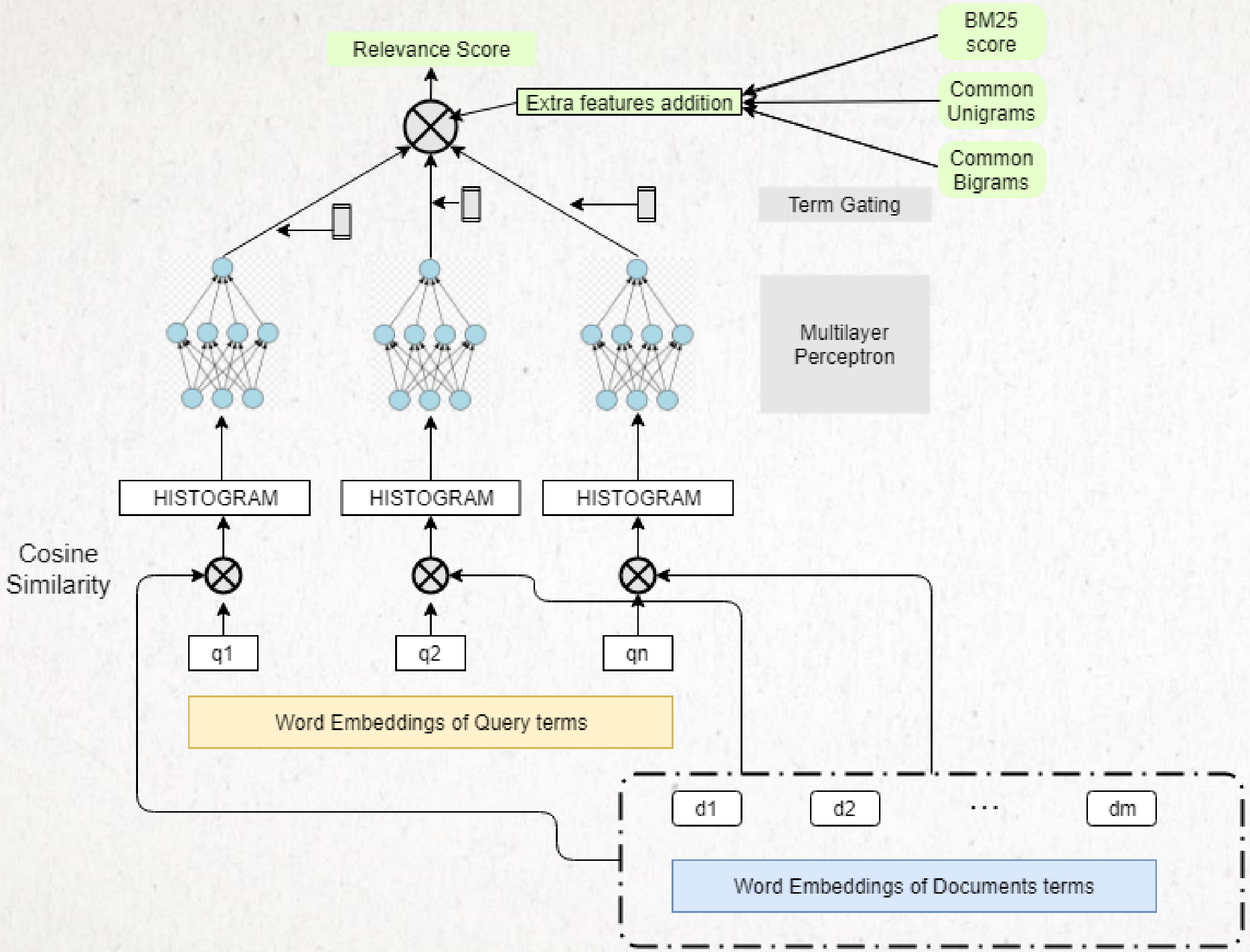
Candidate

Job id: score(4)
Job id: score(2)
Job id: score(5)
...

Job ID

Title
Function
Description
Requirement Summary
Keywords

Deep Relevance Matching Model (DRMM)



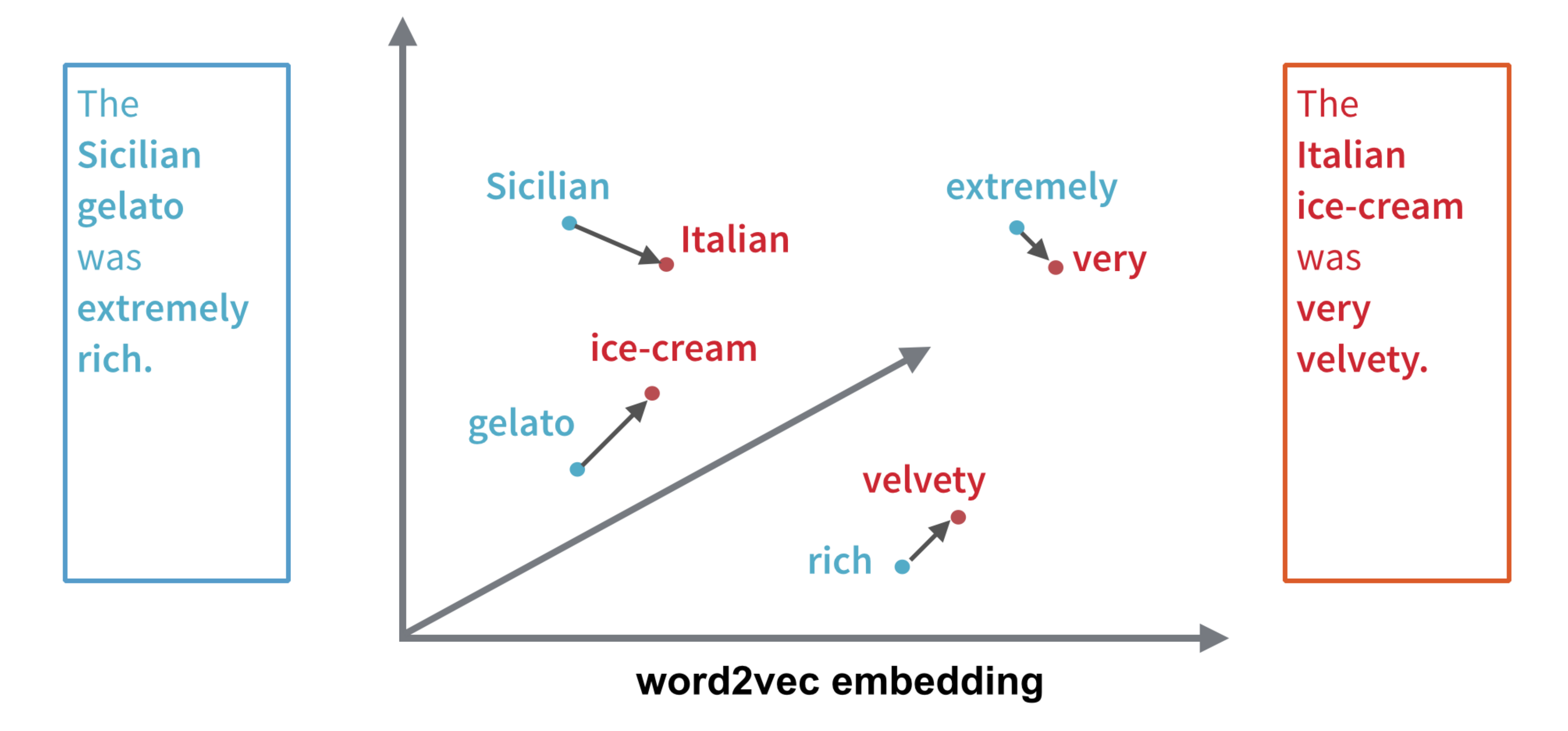
Hinge loss:

Pairwise Ranking Loss:
Separates the positive from the negative documents by scoring the positive higher.

$$L(q, d^+, d^-) = max(0, 1 - s(q, d^+) + s(q, d^-))$$

Word Mover's Distance

Definition: The minimal cumulative distance that the words of the first document need to travel to reach the words of the second document.



Data Preprocessing

- Extract Positive – Negative Pairs
  - Used during training and evaluation on calculation of accuracy
- Create data for MAP experiment
  - Used for calculation of Mean Average Precision

Query Extraction

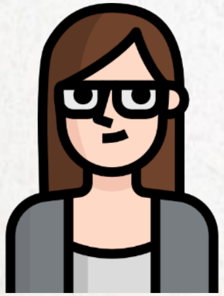
- Maximum query terms: 300
- Use jobs from job list of each candidate except the job with highest score.
- Discard Job not relevant to the rest.
- From each job id:
  - Title
  - Function
  - Keywords
  - Requirement Summary
- Discard Stopwords,
- Discard words on requirement summary with small idf.

DRMM Pairs Data Preprocessing

1. **Query Extraction**
2. Positive Document Extraction
3. Negative Document Extraction
4. DSL Query formation

Positive Document Extraction:

- Bag of words from the highest scored job.
- Search using query with Elasticsearch
- Discard the candidate if positive job not on the results.
- Example:



Customer Service(score 6)

Data Visualization Specialist (score 2)

Business Analyst(score 1)

- Calculate histogram, count of common unigrams and bigrams.

DRMM Pairs Data Preprocessing

1. Query Extraction
2. **Positive Document Extraction**
3. Negative Document Extraction
4. DSL Query formation

Negative Document Extraction:

- Create query bow using 1/6 of the original query.
- Query into Elasticsearch
- Get a document randomly along with BM25
- Calculate histogram, count of common unigrams and bigrams.

DRMM Pairs Data Preprocessing

1. Query Extraction
2. Positive Document Extraction
3. **Negative Document Extraction**
4. DSL Query formation

Domain Specific Language Query for Elasticsearch:

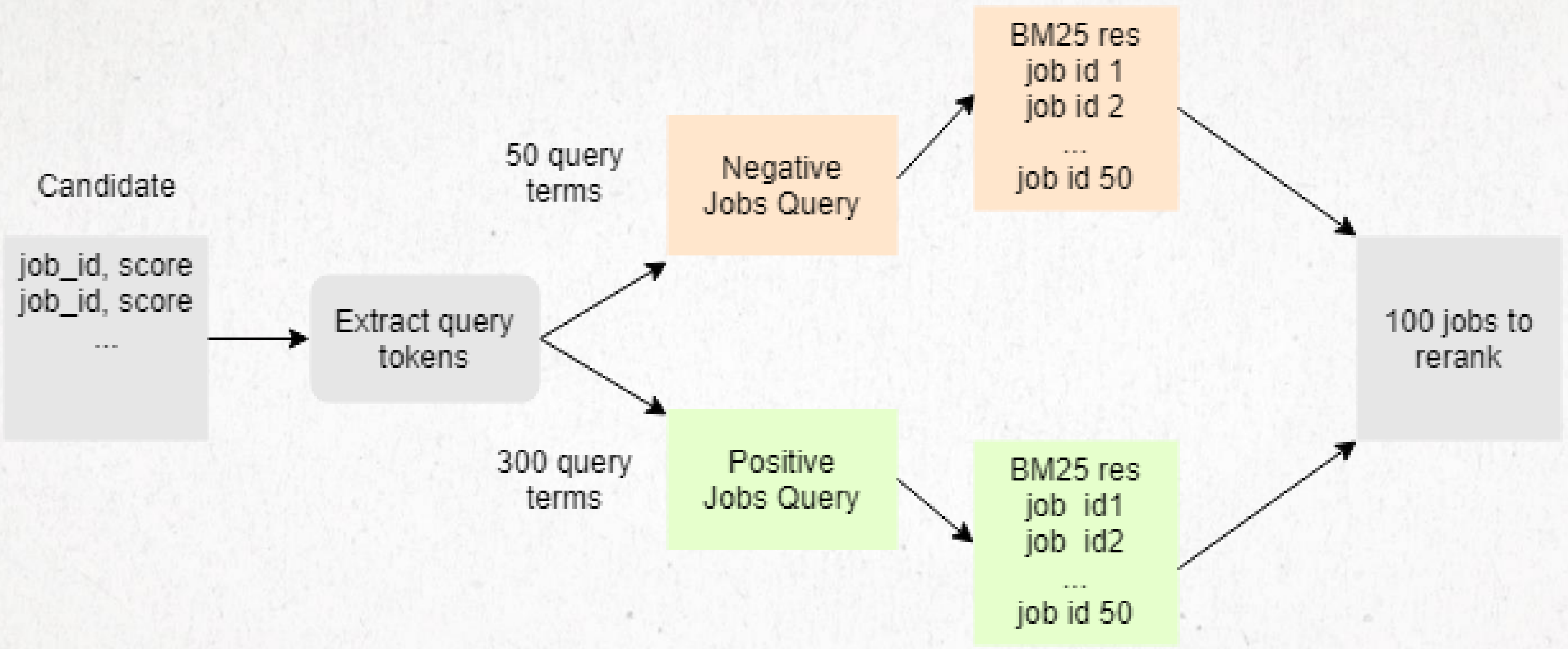
```
GET jobs/_doc/_search
{"query": {
            "bool": {
                "should": [
                    {
                        "multi_match": {
                            "query": query,
                            "type": "most_fields",
                            "fields": ["function^1.5", "title^1.5", "requirement_summary^1.2",
                            "keywords^1.5", "description^1"],
                            "auto_generate_synonyms_phrase_query": "false"

                        }
                    }]
    }
  }
}
```

- Multimatch command(match query to all the fields)
- Should command
- Higher weights on function title and keywords.

DRMM Pairs Data Preprocessing

1. Query Extraction
2. Positive Document Extraction
3. Negative Document Extraction
4. **DSL Query formation**

Preprocessing for Mean Average Precision



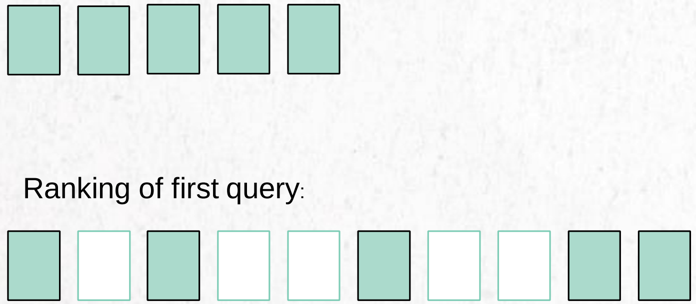
Evaluation Metrics 1

- Accuracy
- Overfitting Metric: Average number of common job ids between a DRMM and Elasticsearch ranking.
- Mean Average Precision at 10

Query 1

Relative documents:

Ranking of first query:

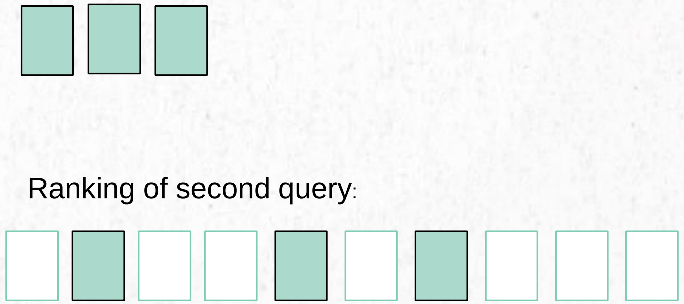


Precision 1 0.5 0.67 0.5 0.4 0.5 0.43 0.38 0.44 0.5

Query 2

Relative documents:

Ranking of second query:



0.0 0.5 0.33 0.25 0.4 0.33 0.43 0.38 0.33 0.3

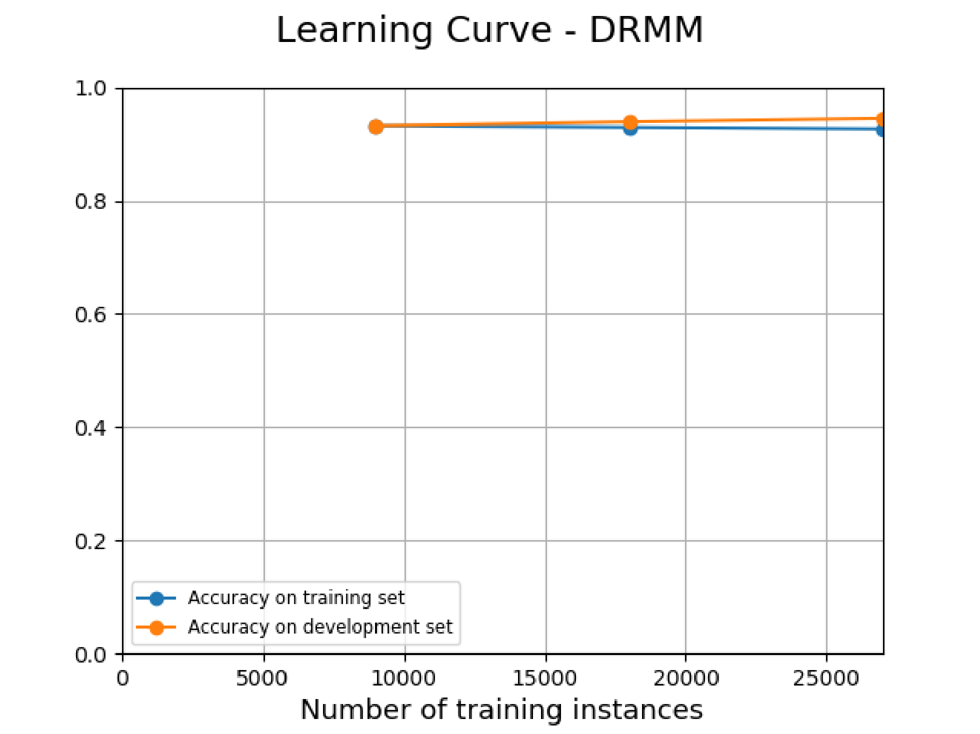
Average Precision Query 1 = (1 + 0.67 + 0.5 + 0.44 +0.5) / 5 = 0.62
Average Precision Query 2 = (0.5 + 0.4 + 0.43) / 3 = 0.44
Mean Average Precision = (0.62 + 0.44) / 2 = **0.53**

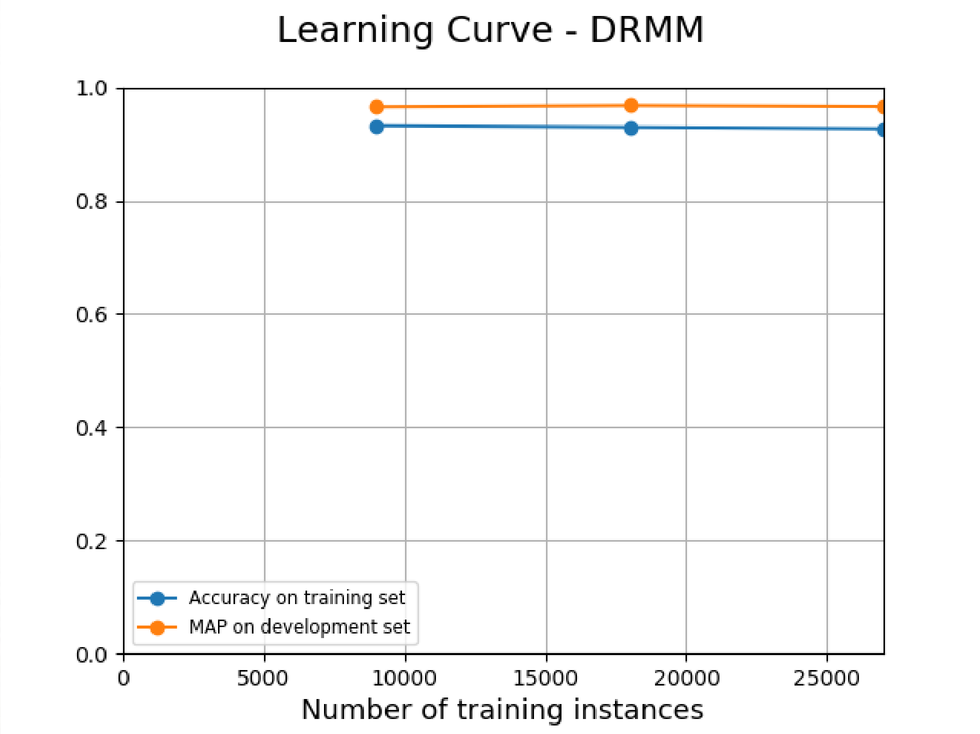
DRMM Training

- **Training settings: 10 hidden layers, 10 nodes per layers, early stop: 0.008.**
- **Version 1**
  - Features : Histograms
- **Version 2**
  - Features : Histograms, count of common unigrams and bigrams, BM25 score
  - Dropout on BM25 = 0.3
  - Overfitting issue.
- **Version 3**
  - Features : Histograms, count of common unigrams and bigrams.
  - Train weights of the MLP first.
  - Freeze MLP and train last layer.
  - Low accuracy and MAP

DRMM Version 1 results – Development Set

num_of_training_data	train_accuracy	accuracy_on_development	map_on_development
9000	0.932111	0.932	0.965704
18000	0.929333	0.939	0.968045
27000	0.926667	0.945	0.966276





DRMM Version 1 results –Test Set

Accuracy_on_test_set	MAP_on_test_set	Baseline MAP on test set(WMD)	Overfitting metric
0.946	0.965532	0.713579	5.637

Comments:

- DRMM can successfully separate the positive job from the negative. (94.6% accuracy)
- It finds the positive documents among the negative in the list (MAP: 96.5%)
- It produces relatively different ranking than BM25 metric(Overfitting metric:5.6)
- The baseline model which is not a random metric in many cases fails.


Thank you