# LEARNING TO RANK

GROUP **3**

Anna Dymanus, Vitor Faria de Souza, Vincent Nguyen,
Nil Palau and Álvaro Sánchez

01

PROBLEM DESCRIPTION

02

LEARNING TO RANK

03

DATA

04

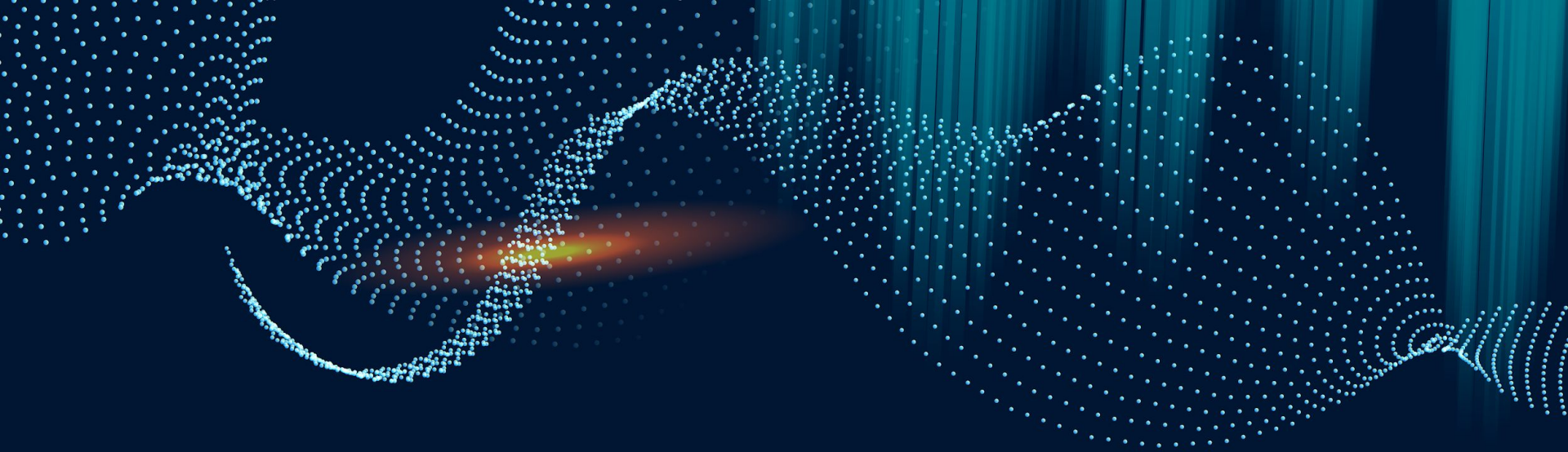
MODELS

05

SCORING AND RESULTS

06

CONCLUSIONS

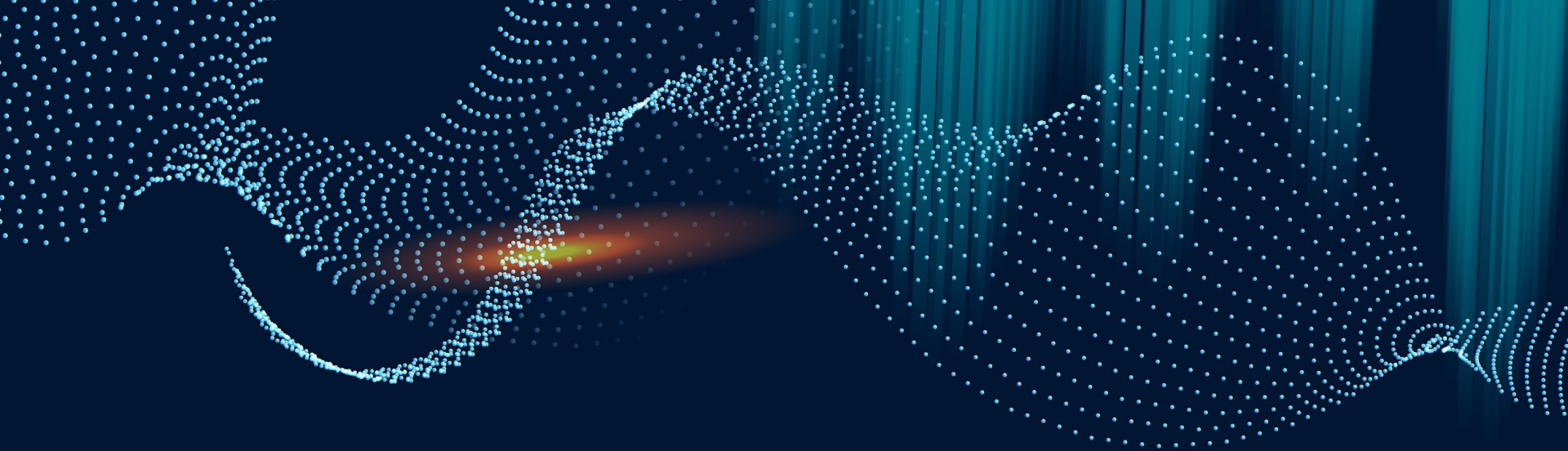


# 01 PROBLEM DESCRIPTION

TREC 2020

# TREC 2020

- For Information retrieval in a *large training data* regime
- One of the few real-world simulated scenarios
- One positive label for 10.000-100.000's per training query
- Ms-marco (Microsoft):
  - **+22** GB (all of it)
- Goal: To study what methods work best in this regime with two different models, point-wise & pair-wise.
- TREC consists of two parts:
  - Passage retrieval task
  - **Document retrieval task**



# 02 LEARNING TO RANK

Point-wise & Pair-wise

# LEARNING TO RANK: DIFFERENT APPROACHES

## POINTWISE

Unsupervised functions

(q, d1)

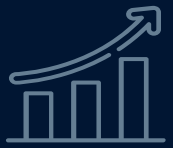
Order by classifier’s confidence

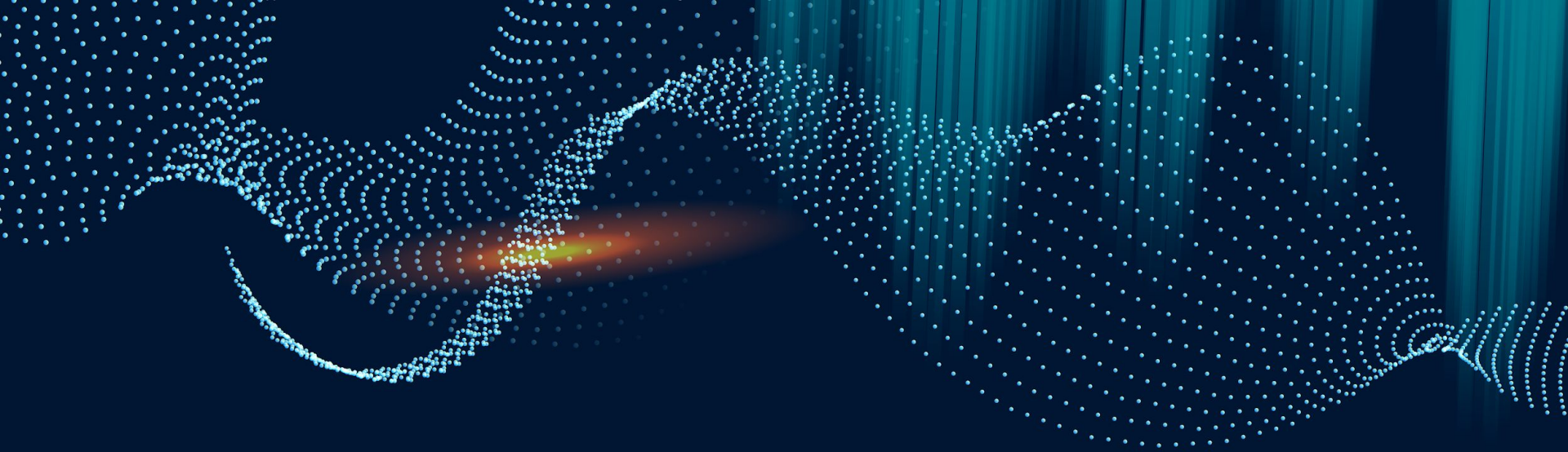
## PAIRWISE

Unsupervised functions

(q, d1, d2)

Merge results in a consistent order





# 03 THE DATA & PREPROCESSING

How we treated the data

# DATA AND DATA PREPROCESSING

DOCUMENT COLLECTION : 3,2 Mio. Documents

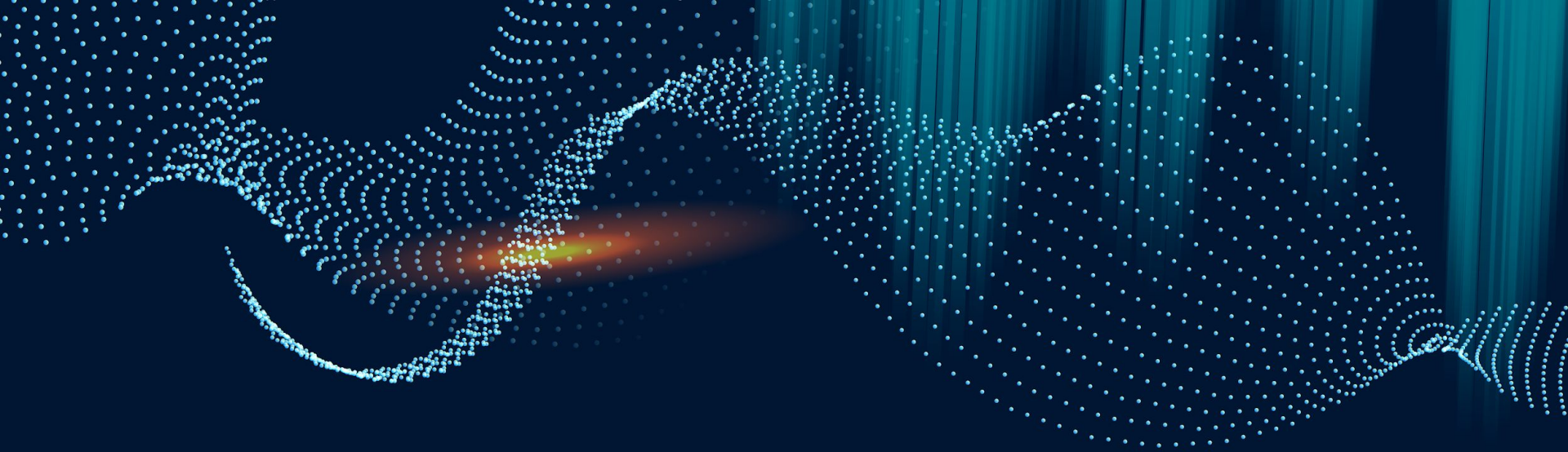
TRAINING SET: ~350 000 queries, each paired with  1 relevant document from the corpus (Label : 1)

VALIDATION SET: ~5 500 queries

TEST SET: ~50 queries, each with ~ 200-1400 documents with relevance labels from 0 to 3

**PREPARE DATASETS**:

- **Problem**: Training set has  only 1 positive example of document per query
- **Solution**: Create 1 negative example of document for each query
    - Assign random other document from the corpus as negative example.
- **PREPROCESS:** Remove stop words, tokenize and lemmatize (spaCy) . Create BoW and Tf-Idf Vector, compress with SVD ...



# 04 THE MODELS

Models for point-wise & pair-wise

# L2R MODELS - MODEL COMPARISON

**BASELINE**: Cosine Similarity

**MODELS WITH DIFFERENT REPRESENTATIONS AS AN INPUT:**

- Tf-idf vector
- Average GloVe Word Embedding
- Average BART Word Embeddings
- BART Document Embedding

**Some Scoring Functions get BoW**

- Jaccard
- BM25

→ BoW served to model together with the representations

# SCORING FUNCTIONS

- Cosine Similarity
- Logistic Regression*
- BM25
- Jaccard Similarity

*(not used for models using BART Embeddings)

# EMBEDDINGS

- **GloVe Embeddings:**
  Non-Contextualised
  Average Word Embedding per Query and Document

- **BART Embeddings:**
  Contextualised
  Takes up to 1024 tokens input
  Average Word Embedding per Query and Document
  Document Embeddings: End of Sequence <EOS>

  **PROBLEM**: Size. After saving up to 35GB

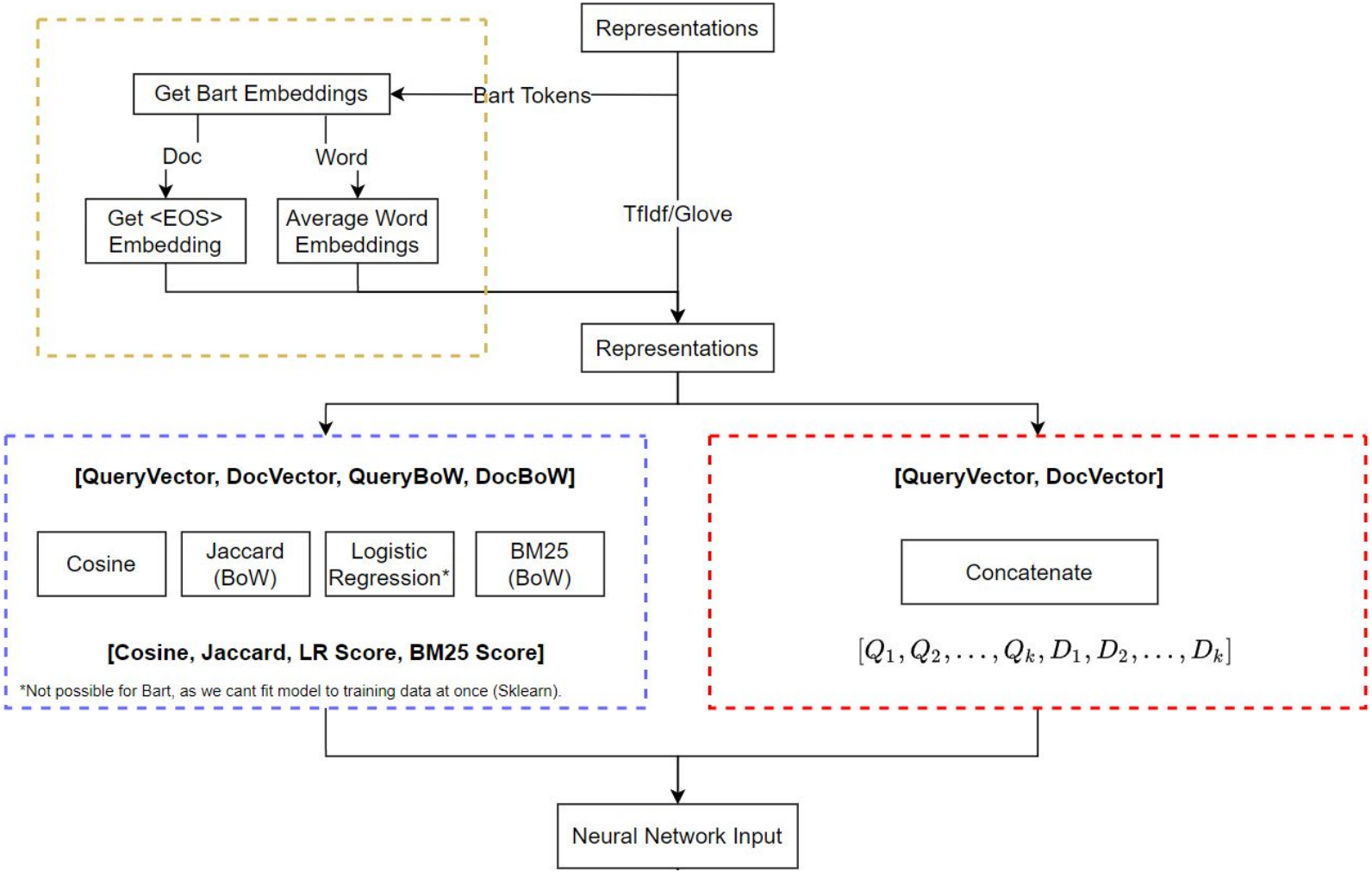
  **SOLUTION**: We don‘t save the embeddings earlier (~35GB)
  BART embeddings calculate during training and testing:
  - Long training runtime!

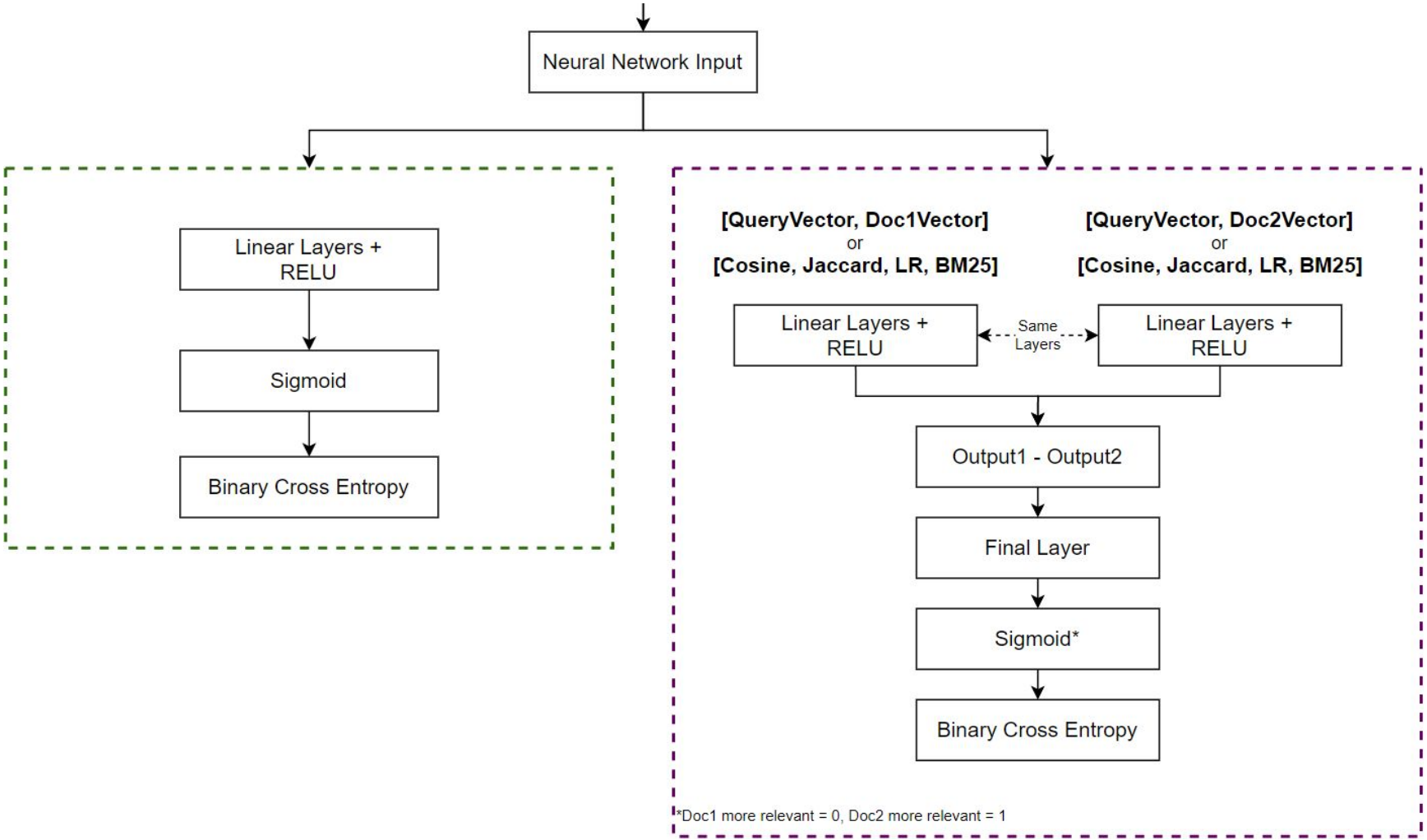
# NEURAL NETWORK TRAINING

SET UP:

- Scoring Mode: *True / False*
- Mode for BART-Embeddings: *None, Word, Document*

- Criterion: *Binary Cross Entropy Loss*
  - Pointwise: Relevant vs. Non-Relevant
  - Pairwise: First vs. Second Document
- Optimizer: *AdamW*
- Scoring Mode: *False*
  - Concatenate Vectors of Query and Document Representations


Representations
Get Bart Embeddings
Bart Tokens
Doc
Word
Get <EOS> Embedding
Average Word Embeddings
Tfldf/Glove
Representations
[QueryVector, DocVector, QueryBoW, DocBoW]
Cosine
Jaccard (BoW)
Logistic Regression*
BM25 (BoW)
[Cosine, Jaccard, LR Score, BM25 Score]
*Not possible for Bart, as we cant fit model to training data at once (Sklearn).
[QueryVector, DocVector]
Concatenate
$[Q_1, Q_2, \ldots, Q_k, D_1, D_2, \ldots, D_k]$
Neural Network Input


Neural Network Input
Linear Layers + RELU
Sigmoid
Binary Cross Entropy
[QueryVector, Doc1Vector] or [Cosine, Jaccard, LR, BM25]
[QueryVector, Doc2Vector] or [Cosine, Jaccard, LR, BM25]
Linear Layers + RELU
Same Layers
Linear Layers + RELU
Output1 - Output2
Final Layer
Sigmoid*
Binary Cross Entropy
*Doc1 more relevant = 0, Doc2 more relevant = 1

# Rank the results: Pointwise vs Pairwise

**Pointwise L2R:**

- Rank the documents based on the model's score.

**Pairwise L2R:**

- **PROBLEM:** Each of ~50 queries has ~200-1400 documents to be ranked → 40 000-2 000 000 comparisons per query: computationally too heavy
  - O(n log n) Sorting: Still up to 15 000 Comparisons per Query.

- **OUR APPROACH:**
  - For each query create a vector with average embeddings of all the documents that are to be ranked.
  - Compare how relevant each of those document is in relation to that average document vector.
  - Base the rank on that relevance score.

# 05

# SCORING AND RESULTS

Obtained results

# Evaluation Metrics

**Gold Standard:**

- DocID and QueryID pairs with relevance ratings ranging from 0 (irrelevant) to 3 (super relevant).

**Mean Reciprocal Rank @ k=10:**

- Position of the first relevant document (rating ≥ 2) among the first 10 retrieved.

**(normalized) Discounted Cumulative Gain:**

- How close the ranked list of retrieved documents is to the perfect order.

# MRR Relevance Threshold = 1

| | | MRR_at_10 | DCG | nDCG | pairwise_acc |
|---|---|---|---|---|---|
| (Pointwise) | GloVe Word Embedding | 0.568309 | 32.664117 | 0.676987 | 0.607968 |
| (Pairwise) | GloVe Word Embedding Scoring Pairwise | 0.572250 | 32.295990 | 0.668845 | 0.595777 |
| (Pointwise) | Tf-Idf | 0.481423 | 31.800888 | 0.667206 | 0.592718 |
| (Pairwise) | Tf-Idf Pairwise | 0.564212 | 31.706860 | 0.666408 | 0.583277 |
| (Pairwise) | GloVe Word Embedding Pairwise | 0.581654 | 32.100947 | 0.665090 | 0.587289 |
| (Pointwise) | BART Doc Embedding Scoring | 0.456966 | 36.038576 | 0.664488 | 0.574078 |
| (Pointwise) | GloVe Word Embedding Scoring | 0.537302 | 32.126986 | 0.663315 | 0.594484 |
| (Pointwise) | Tf-Idf Scoring | 0.451477 | 31.722231 | 0.655401 | 0.587863 |
| (Pairwise) | Tf-Idf Scoring Pairwise | 0.448247 | 31.595256 | 0.651772 | 0.583725 |
| (Pointwise) | BART Doc Embedding | 0.520378 | 33.681145 | 0.650810 | 0.549932 |
| | Cosine Similarity | 0.376966 | 31.337178 | 0.649092 | 0.581657 |
| (Pairwise) | BART Doc Embedding Pairwise | 0.521152 | 31.146306 | 0.646021 | 0.555442 |
| (Pointwise) | BART Word Embedding Scoring | 0.473348 | 31.333809 | 0.643832 | 0.567777 |
| (Pairwise) | BART Word Embedding Pairwise | 0.491113 | 31.226614 | 0.641721 | 0.561485 |
| (Pointwise) | BART Word Embedding | 0.470986 | 31.208557 | 0.633229 | 0.556830 |

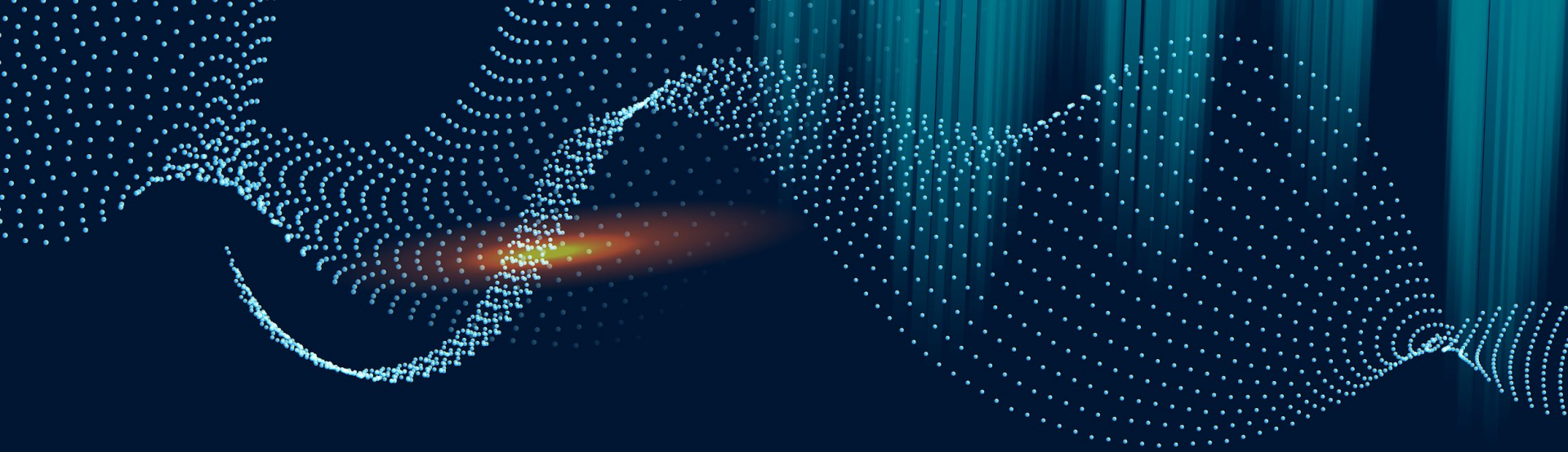
Best model

Baseline

# MRR Relevance Threshold = 2

| | | MRR_at_10 | DCG | nDCG | pairwise_acc | |
|---|---|---|---|---|---|---|
| (Pointwise) | GloVe Word Embedding | 0.380159 | 32.664117 | 0.676987 | 0.607968 | Best model |
| (Pairwise) | GloVe Word Embedding Scoring Pairwise | 0.331294 | 32.295990 | 0.668845 | 0.595777 | |
| (Pointwise) | Tf-Idf | 0.251800 | 31.800888 | 0.667206 | 0.592718 | |
| (Pairwise) | Tf-Idf Pairwise | 0.252473 | 31.706860 | 0.666408 | 0.583277 | |
| (Pairwise) | GloVe Word Embedding Pairwise | 0.318217 | 32.100947 | 0.665090 | 0.587289 | |
| (Pointwise) | BART Doc Embedding Scoring | 0.231349 | 36.038576 | 0.664488 | 0.574078 | |
| (Pointwise) | GloVe Word Embedding Scoring | 0.303027 | 32.126986 | 0.663315 | 0.594484 | |
| (Pointwise) | Tf-Idf Scoring | 0.258869 | 31.722231 | 0.655401 | 0.587863 | |
| (Pairwise) | Tf-Idf Scoring Pairwise | 0.237495 | 31.595256 | 0.651772 | 0.583725 | |
| (Pointwise) | BART Doc Embedding | 0.210084 | 33.681145 | 0.650810 | 0.549932 | |
| | Cosine Similarity | 0.253387 | 31.337178 | 0.649092 | 0.581657 | Baseline |
| (Pairwise) | BART Doc Embedding Pairwise | 0.203092 | 31.146306 | 0.646021 | 0.555442 | |
| (Pointwise) | BART Word Embedding Scoring | 0.280131 | 31.333809 | 0.643832 | 0.567777 | |
| (Pairwise) | BART Word Embedding Pairwise | 0.216584 | 31.226614 | 0.641721 | 0.561485 | |
| (Pointwise) | BART Word Embedding | 0.191528 | 31.208557 | 0.633229 | 0.556830 | |

| GloVe Pointwise Model - Examples | MRR_AT_10 | nDCG |
|---|---|---|
| HARDEST QUERIES | | |
| **what is theraderm used for** | 0.0 | 0.256076 |
| **how many liberty ships were built in brunswick** | 0.0 | 0.261792 |
| **what is an aml surveillance analyst** | 0.0 | 0.370303 |
| EASIEST QUERIES | | |
| **what is durable medical equipment consist of** | 1.0 | 0.895841 |
| **exons definition biology** | 0.5 | 0.859406 |
| **define visceral?** | 0.5 | 0.867105 |



# 06 CONCLUSIONS

**What have we learned from our results**

# Conclusions

- Transformers Embeddings (e.g. BART) are costly to get
  - With limited time → worst
  - Not effective without fine-tuning
- GloVe Embeddings work best and are fast to get (just a lookup)
- Embedding vs Scoring:
  - Embedding directly as input is better
- Pairwise has a slightly worse performance than Pointwise, even has more difficulties:
  - Needs more GPU memory (2 docs per data-point)
  - Ranking the results is costly if done properly
- Plenty space for improvement: Hyperparameter Optimization, Ensembles etc.
  - Achieving best possible results was not our focus.
  - Comparing representations was.

# THANKS!

Do you have any questions?

Our code: https://github.com/annadymanus/IR-project

Anna Dymanus
Vitor Faria de Souza
Vincent Nguyen
Nil Palau
Álvaro Sánchez