



# Query Performance Prediction: Techniques and Applications in Modern Information Retrieval

Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi & Ebrahim Bagheri

Tutorial at the 2nd International ACM SIGIR Conference on Information Retrieval in the Asia Pacific  
(SIGIR-AP 2024)

December, 2024



UNIVERSITY OF AMSTERDAM



# Presenters



**Negar Arabzadeh**  
University of Waterloo  
Canada



**Chuan Meng**  
University of Amsterdam  
The Netherlands



**Mohammad Aliannejadi**  
University of Amsterdam  
The Netherlands



**Ebrahim Bagheri**  
Toronto Metropolitan University  
Canada

# Supplementary materials

We offer the implementation of a collection of pre- and post-retrieval QPP methods in Python and PyTorch framework.



QR code for the repo

# Overview

1. What is QPP and Why QPP
2. Pre-retrieval QPP
3. Post-retrieval QPP
4. Break
5. QPP for various search scenarios
6. Applications of QPP
7. Conclusions and future directions
8. Discussion

# Query Performance Prediction

Query performance prediction (QPP), a.k.a.  
query difficulty prediction is  
to predict the retrieval quality of a search system for a  
query *without human relevance judgments.*

## What is a difficult query?

- Poorly-performed
  - E.g., performance < specific threshold  $T$

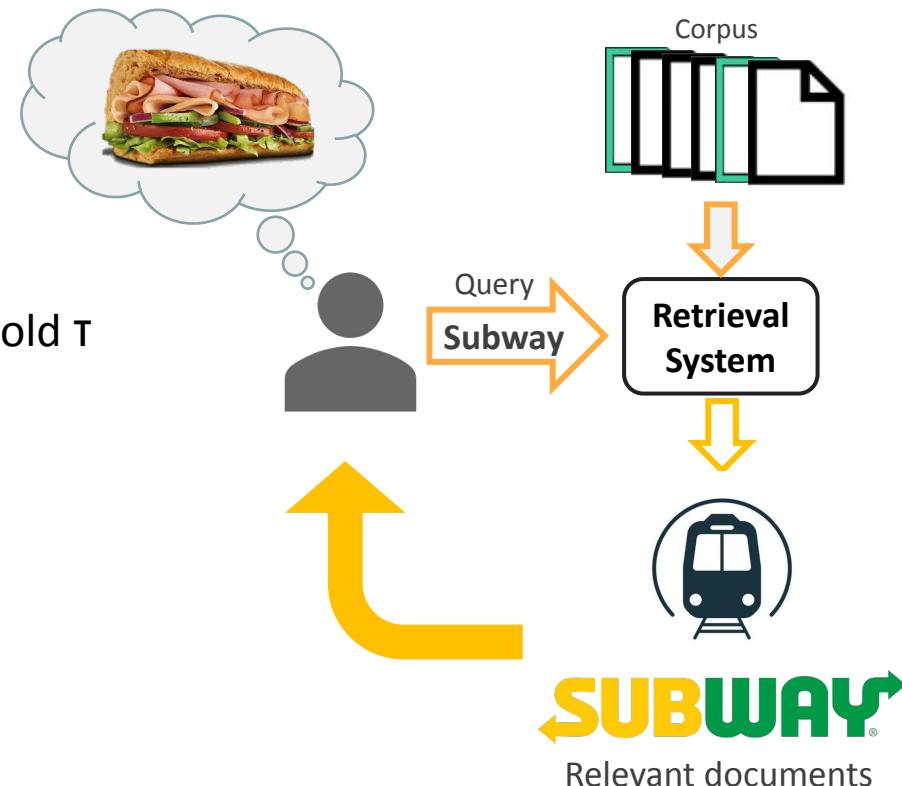
# Introduction

## What is a difficult query?

- Poorly-performed
  - E.g., performance < specific threshold  $T$

## Why is a query “difficult”?

1. Query term ambiguity

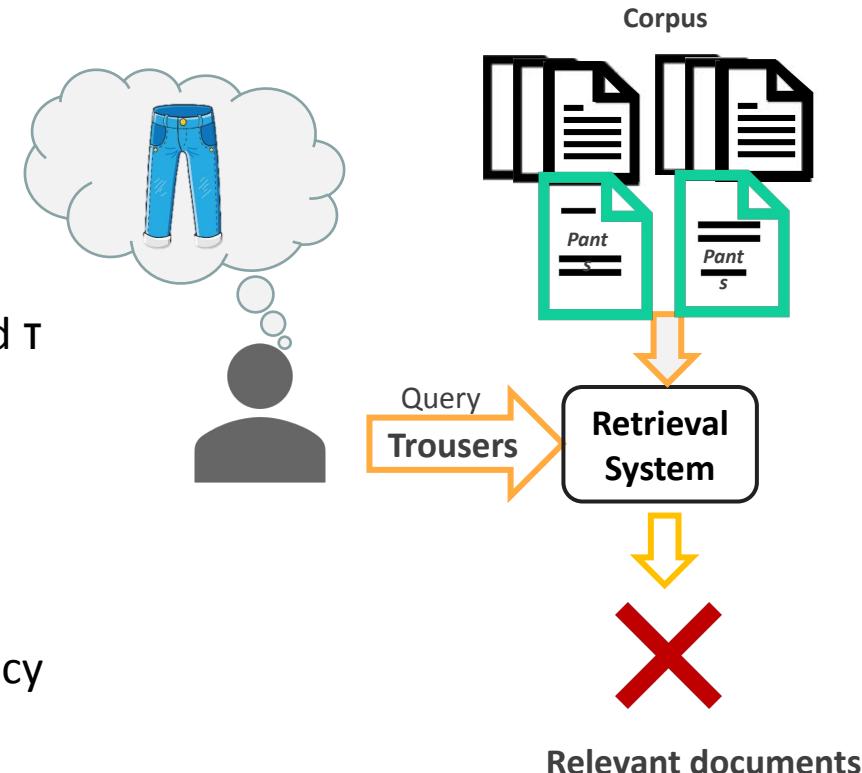


## What is a difficult query?

- Poorly-performed
  - E.g., performance < specific threshold  $T$

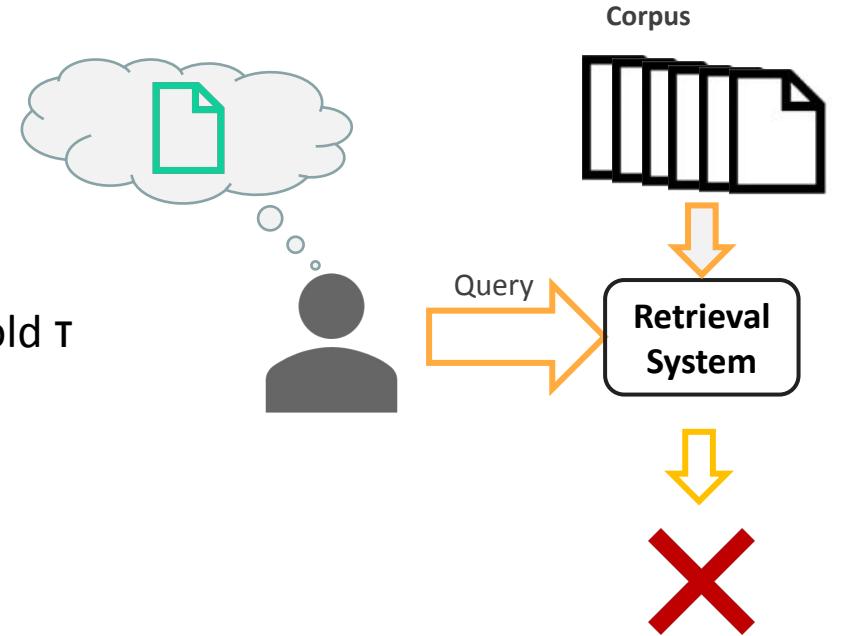
## Why is a query “difficult”?

1. Query term ambiguity
2. Query and document language inconsistency



## What is a difficult query?

- Poorly-performed
  - E.g., performance < specific threshold  $T$



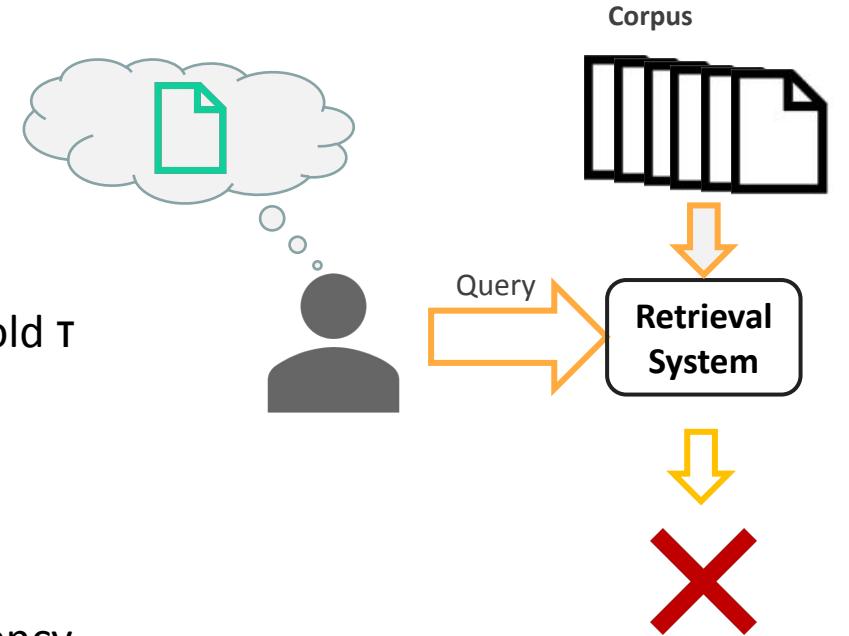
## Why is a query “difficult”?

1. Query term ambiguity
2. Query and document language inconsistency
3. Lack of relevant documents

Relevant documents

## What is a difficult query?

- Poorly-performed
  - E.g., performance < specific threshold  $T$



## Why is a query “difficult”?

1. Query term ambiguity
2. Query and document language inconsistency
3. Lack of relevant documents
4. Among others!

Relevant documents

## What is a **difficult** query?

- Poorly-performed one

## Why is a query **difficult** ?

- Different reasons

# Introduction

## What is a **difficult** query?

- Poorly-performed one

## Why is a query **difficult** ?

- Different reasons

**The goal:** Estimating the performance of individual queries so we can further address the hard-to-satisfy queries better.

# Problem definition

## Query Performance Problem (QPP)

Predicting the quality of retrieved documents, in satisfying the information needs behind the query.

**Given:**

- A collection D
- A list of retrieved documents  $D_q$
- A query q,

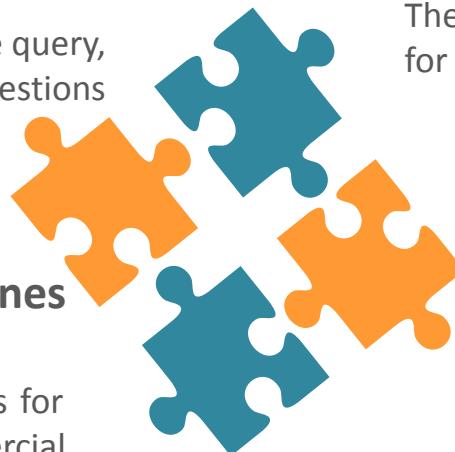
**Predictor  $\mu$  has to estimate the Average Precision of q , AP (q):**

$$\widehat{AP(q)} \leftarrow \mu(q, D_q, D)$$

# Primary Applications

## Feedback to users

The user can rephrase the query,  
e.g., asking clarifying questions



## Feedback to search engines

When there is no relevant documents for the query , especially in commercial search engines, the need to expand the collection for difficult queries is sensed.

## Feedback to system Administrator

The search engine can use different strategies for a different query.

## Information Retrieval administrator

It can help to merge result of a query over different data .

# Primary Applications

Feedback to users

The user expects to receive

**We will revisit the applications in depth later in this tutorial**

Feedback to search engines

When there is no relevant documents for the query , especially in commercial search engines, the need to expand the collection for difficult queries is sensed.



Feedback to system Administrator

The search engine can use different strategies for a different query.

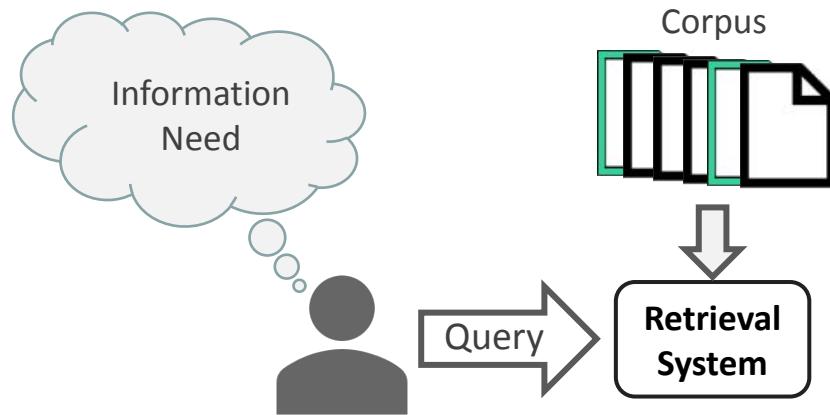
join multiple data sources  
join multiple data sources  
join multiple data sources  
join multiple data sources

It can help to merge result of a query over different data .



# Categories

## Pre-retrieval

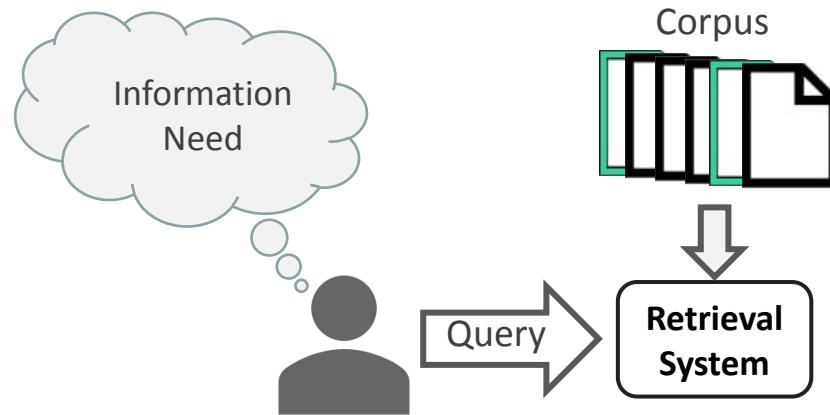


No access to retrieved Items

Is this system going to satisfy the information need of the user?

# Categories

Pre-retrieval

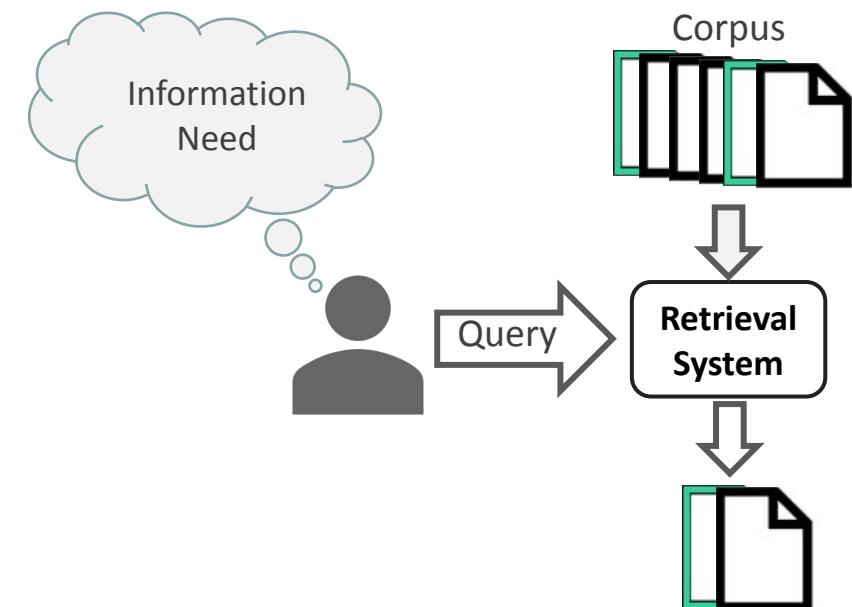


No access to retrieved Items

Is this system going to satisfy the information need of the user?



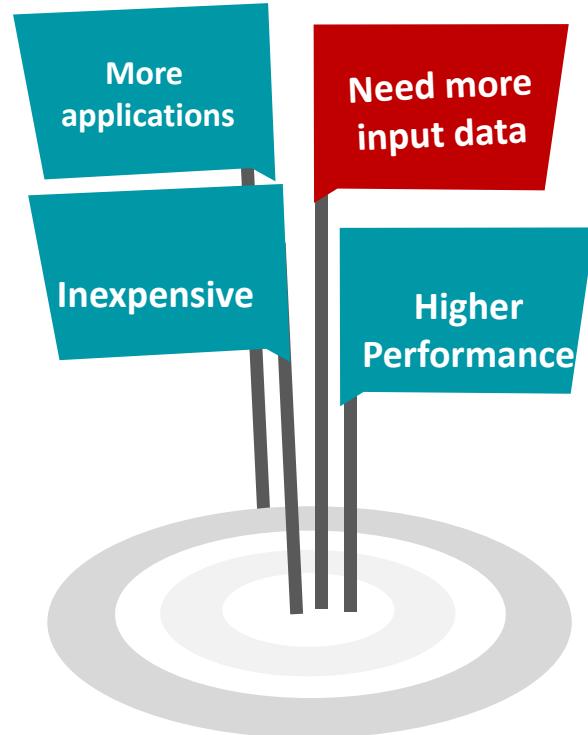
Post-retrieval



How good are the retrieved documents w.r.t satisfying the information need?

# Categories

Pre-retrieval QPP



Post-retrieval QPP

# QPP Evaluation

# QPP evaluation

**Given:**

- A collection D
- A list of retrieved documents  $D_q$
- A query q,

Predictor  $\mu$  has to estimate the Average Precision of q ,  $AP(q)$ :

$$\widehat{AP}(q) \leftarrow \mu(q, D_q, D)$$

## How good is the predicted quality?

$$Quality(\mu) = correlation([AP(q_1) \dots AP(q_n)], [\widehat{AP}(q_1) \dots \widehat{AP}(q_n)])$$

	Actual performance	Predicted Performance
$q_1$	$AP(q_1)$	$AP'(q_1)$
$q_2$	$AP(q_2)$	$AP'(q_2)$
...	...	...
$q_n$	$AP(q_n)$	$AP'(q_n)$

20



# QPP evaluation

Most common evaluation: correlation-based evaluation approaches

- The correlation based evaluation method first mentioned in 1998 [1]
- Correlation between predicted ranking quality and actual ranking quality for a set of queries, in terms of an IR evaluation metrics
- Two widely-used correlation coefficients:
  - Linear: Pearson's  $\rho$
  - Rank-based: Kendall's  $\tau$ , Spearman's  $\rho$

# QPP evaluation

Drawback: correlation-based approaches evaluate QPP at a very high level, summarizing the performance of a QPP method over a set of queries into a single correlation coefficient.

- Faggioli et al. [1] propose two new fine-grained metrics
  - scaled Absolute Rank Error (sARE)

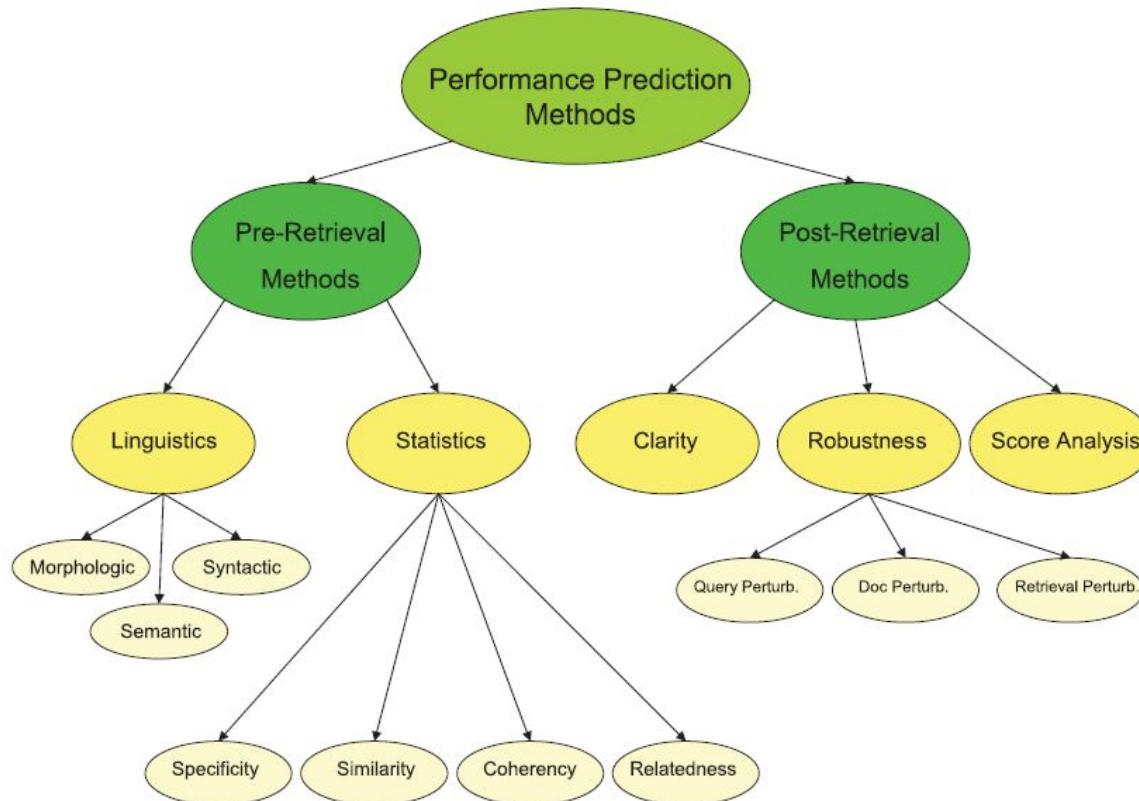
$$\text{sARE-AP}(q_i) := \frac{|r_i^p - r_i^e|}{|Q|},$$

- scaled Mean Absolute Rank Error (sMARE)

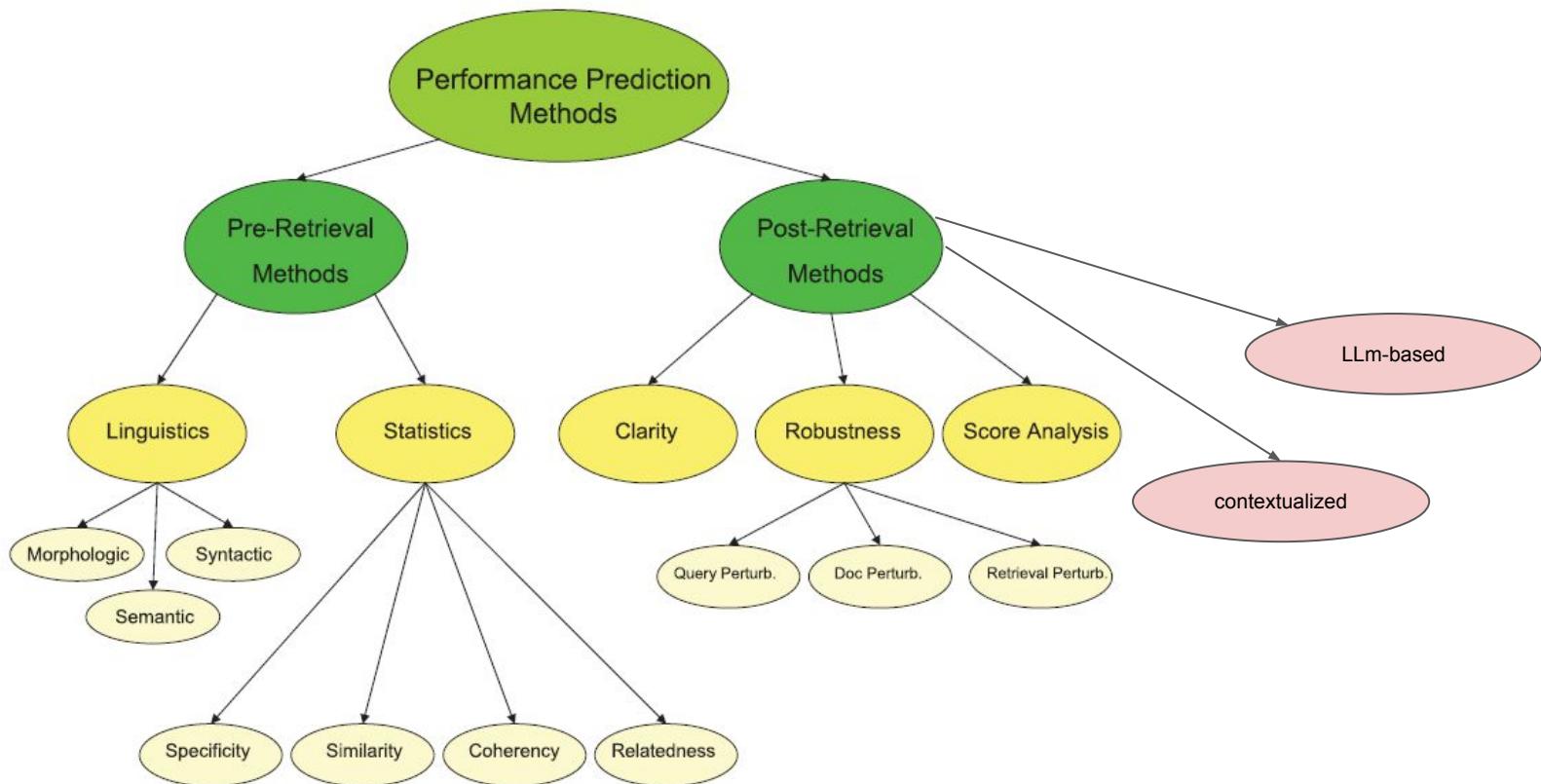
$$\text{sMARE-AP}(\mathcal{P}) := \frac{1}{|Q|} \sum_{q_i \in Q} \text{sARE-AP}(q_i).$$

# QPP Categorization

# Categories



# Categories

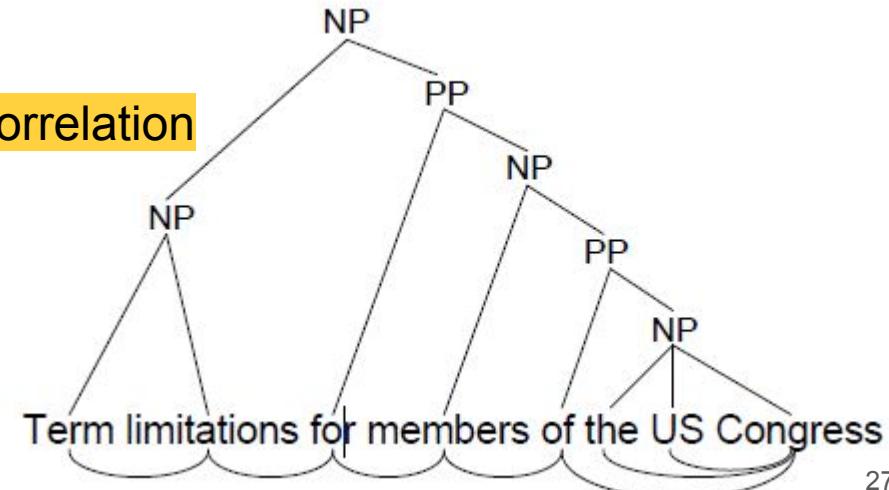


# Pre-retrieval QPP

# Linguistic approaches

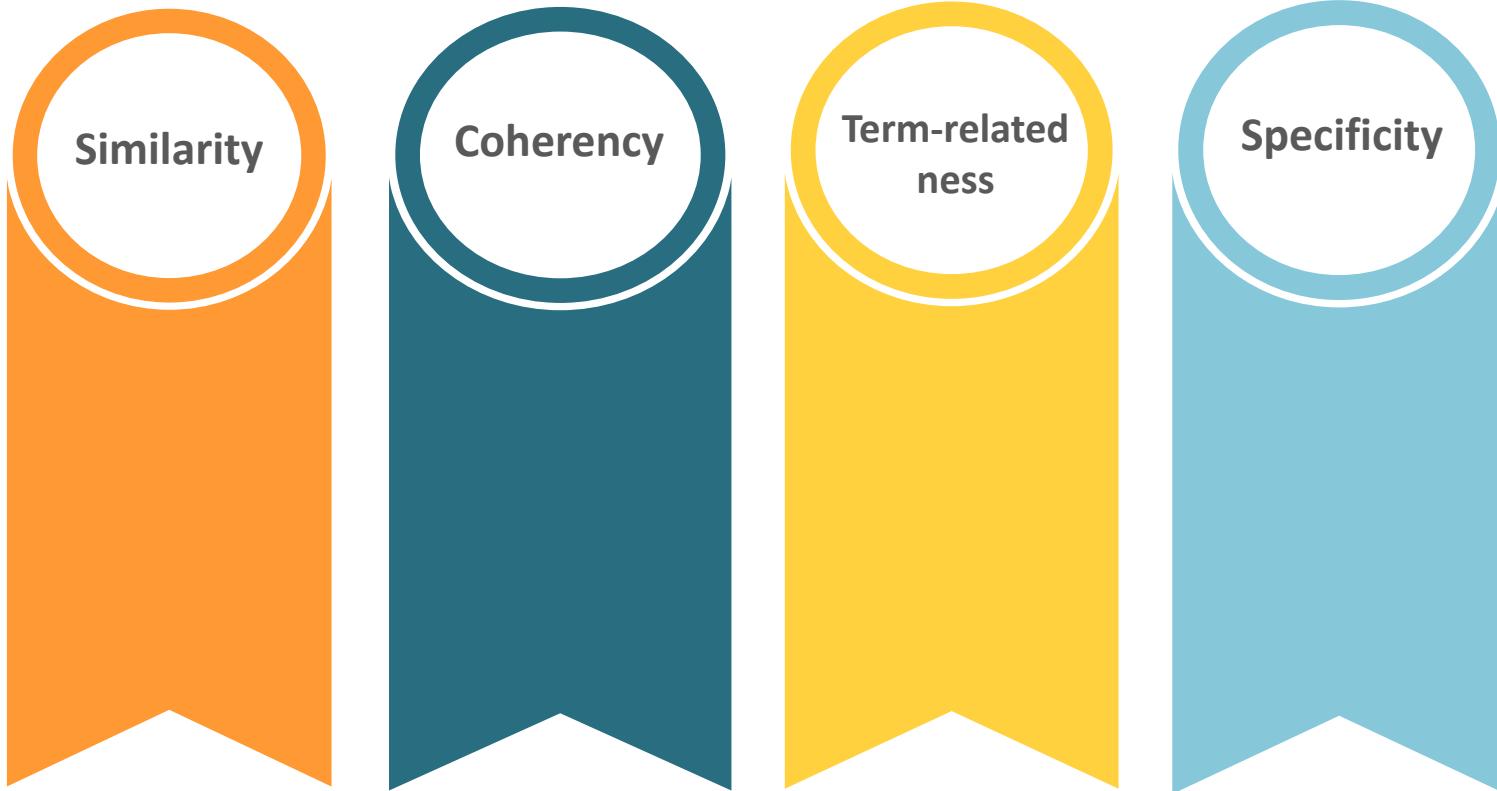
- a. Morphological: Average number of morphemes per query word, presence of proper nouns, acronyms, numeral values, and unknown tokens.
- b. Syntactical: Depth of syntactic parse tree and syntactic link span, indicating grammatical relationships and complexity.
- c. Polysemy Assessment: Utilizes the WordNet database to measure the average number of meanings (synsets) per word.

Most linguistic features showed weak or no correlation with system performance.



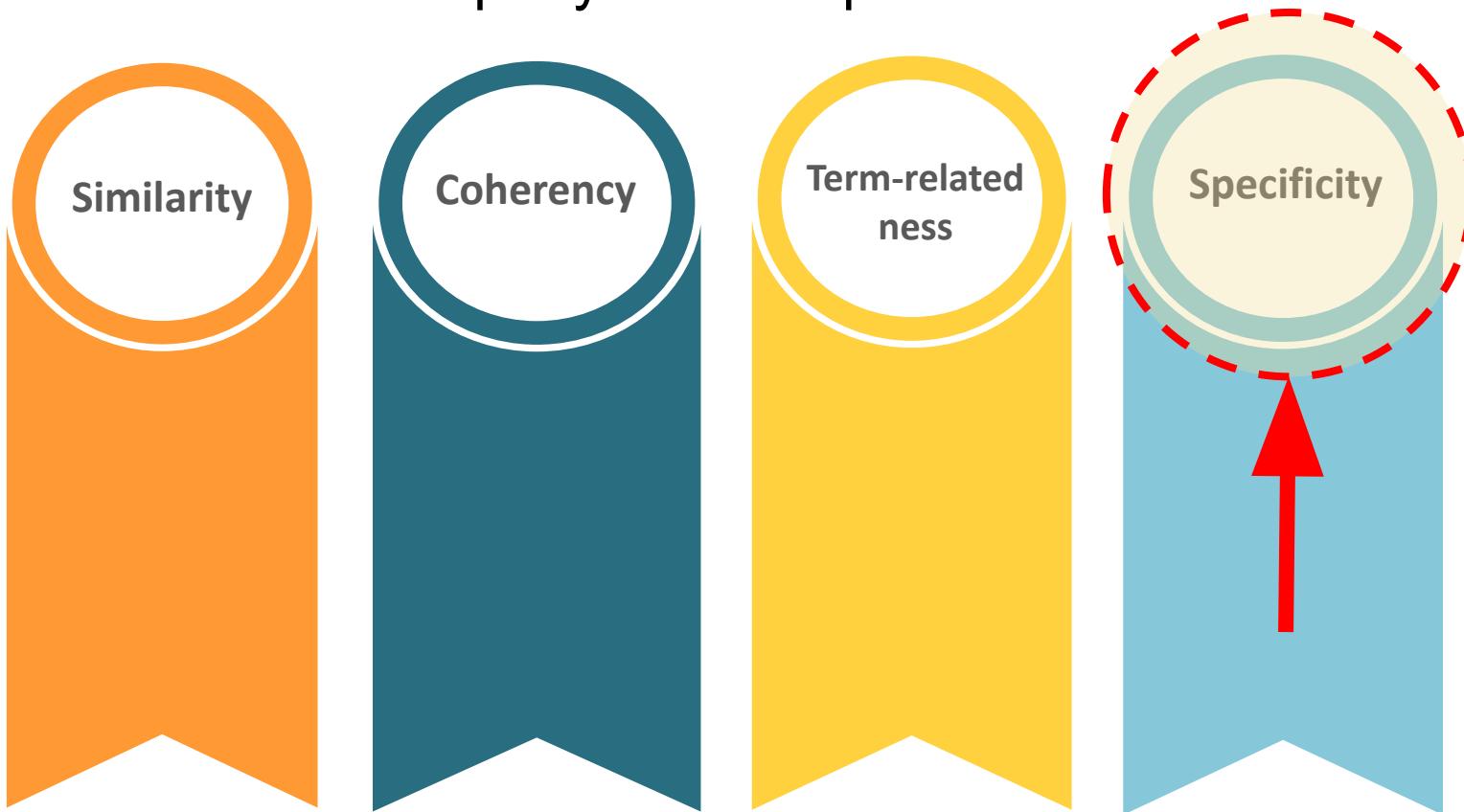
# Statistical approaches

Intuition: Distribution of query term frequencies within the collection



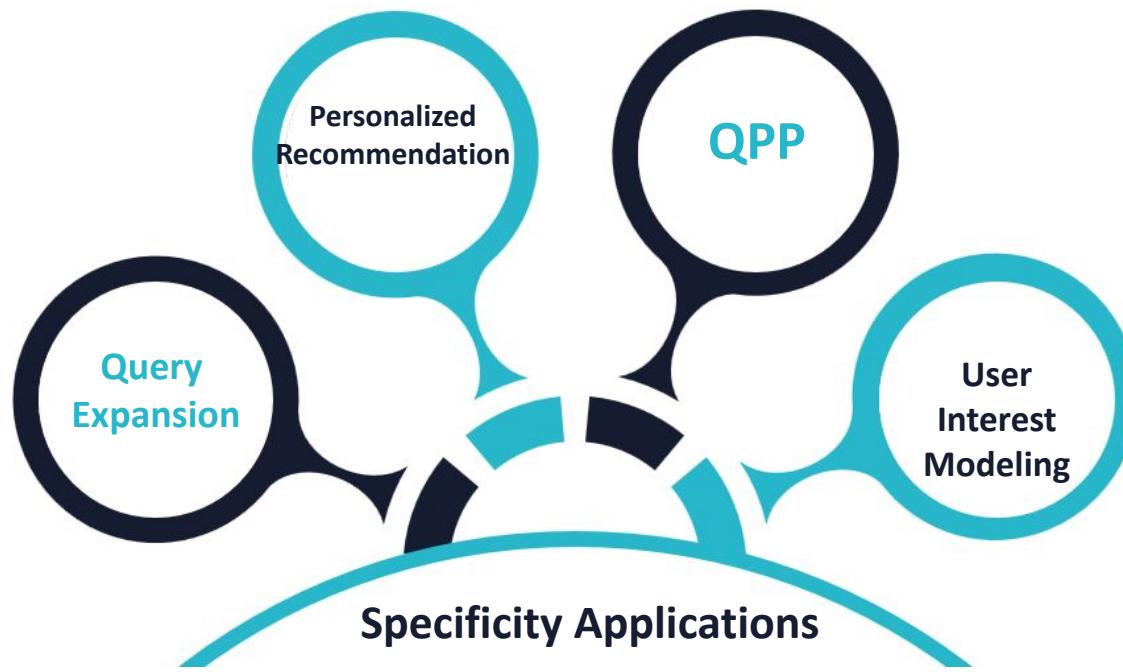
# Statistical approaches

Intuition: Distribution of query term frequencies within the collection



# Specificity

**Specificity Definition:** The level of detail in which a given term is represented



# Specificity-based QPP - IDF

**Idea:** relative importance of the query terms

→ Inverse document frequency (idf):

$$idf(t) = \log\left(\frac{N}{N_t}\right)$$

N : Number of documents in the collection

$N_t$ : Number of documents containing term t

# Specificity-based QPP- ICTF

**Idea:** relative importance of the query terms

- Inverse document frequency (idf):

$$idf(t) = \log \left( \frac{N}{N_t} \right)$$

N : Number of documents in the collection

$N_t$ : Number of documents containing term t

- inverse collection term frequency (ictf)

$$ictf(t) = \log \left( \frac{|D|}{tf(t, D)} \right)$$

$|D|$  is the number of all terms in collection D

$tf(t, D)$  term frequency of term t in D

# Specificity-based QPP - SCS

**Idea:** difference between query and collection language model

**simplified clarity score (SCS):**measures the Kullback-Leibler divergence of the simplified query language model from the collection language model.

$$SCS(q) = \sum_{t \in q} Pr(t|q) \log \left( \frac{Pr(t|q)}{Pr(t|D)} \right).$$

Approximated by maximum likelihood estimation of selecting a term from the language model of the query or collection.

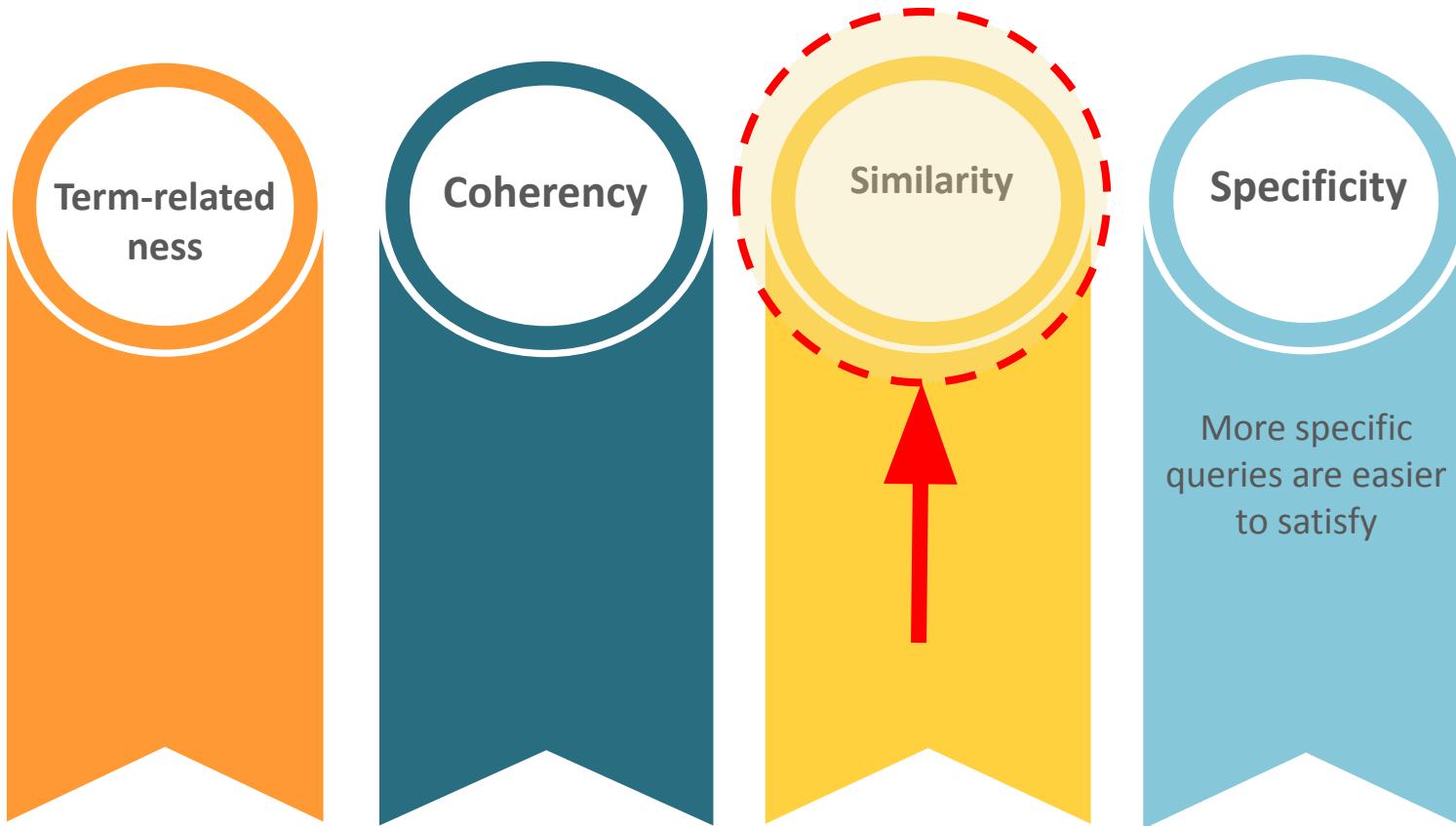
# Specificity-based QPP -QS

**Idea:** ease of separating the relevant and non-relevant document

**Query Scope (QS):** measures the percentage of documents containing at least one of the query terms in the collection.

→ High query scope indicates many candidates for retrieval thus separating relevant results from non-relevant results might be more difficult.

# Statistical approaches



# Similarity-based QPP

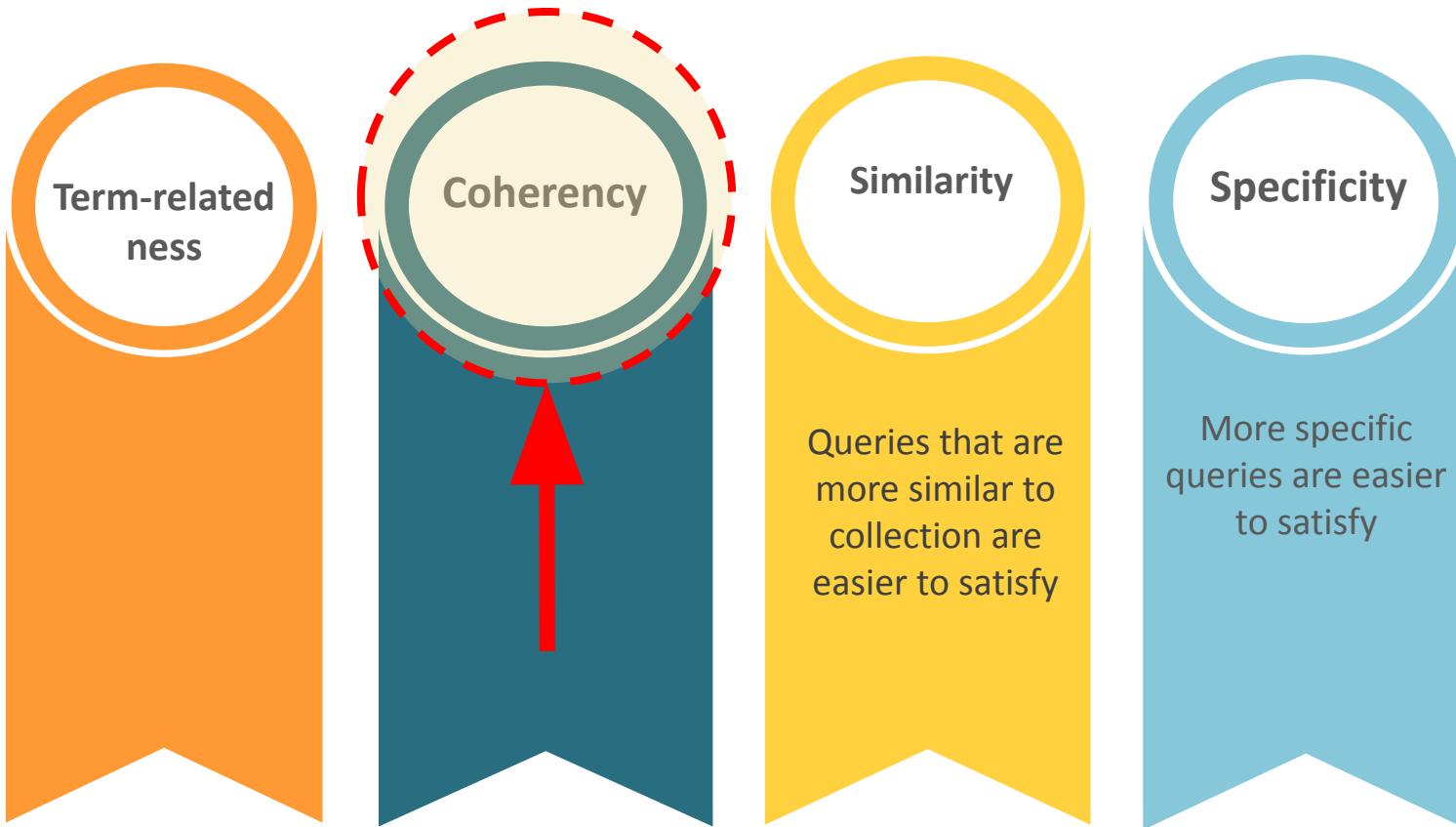
**Idea:** Similarity of the query and collection.

**Similarity of the collection and Query (SCQ):** Queries that are similar to the collection are easier to answer since high similarity potentially indicates the existence of many relevant documents to the Query.

**Approach:** Measuring the vector-space based query similarity to the collection, while considering the collection as a one large document composed of concatenation of all the documents.

$$SCQ(t) = (1 + \log(tf(t, D))) \cdot idf(t)$$

# Statistical approaches



# Coherency-based QPP

**Idea:** Inter-similarity of relevant documents

**Approach:** Associating each term in the with a coherence score reflecting the average pairwise similarity between all pairs of documents containing the term.

**Drawback:** heavy analysis during indexing time

# Coherency-based QPP

**Idea:** Inter-similarity of relevant documents

**Approach:** Associating each term in the with a coherence score reflecting the average pairwise similarity between all pairs of documents containing the term.

**Drawback:** heavy analysis during indexing time

Alternative  $\text{VAR}(t)$ : variance of the term weights over the documents containing it in the collection.

Low variance of the term weight distribution

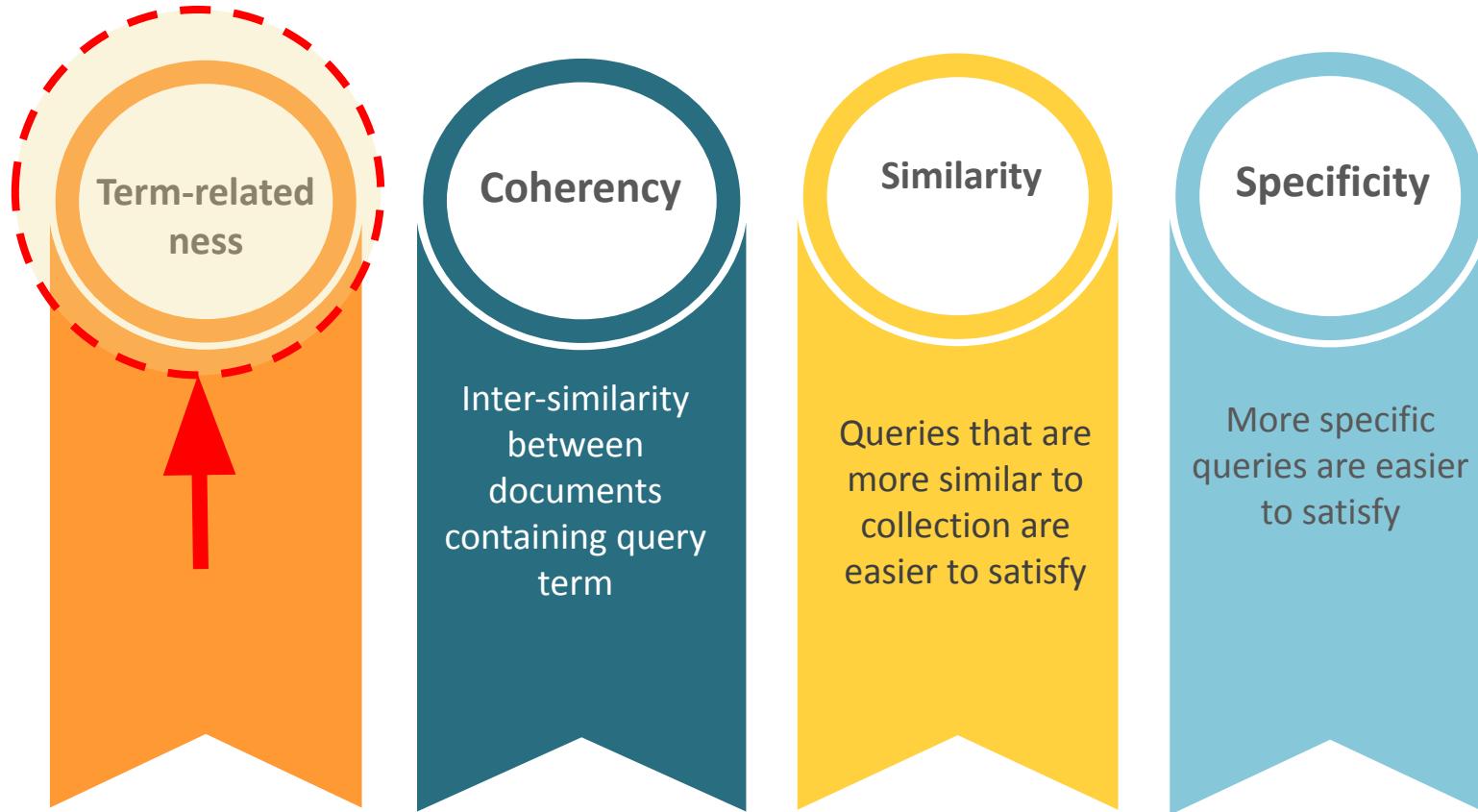


less distinguishability of between highly relevant and less relevant documents



probably more difficult query

# Statistical approaches



# Term relatedness-based QPP

**Idea:** The more the query terms co-occur - the easier it is to satisfy the query  
→ assuming all query terms are related to the same topic.

Example: “high blood pressure”

Pointwise mutual information (PMI) : measure of co-occurrence statistics of two terms in the collection

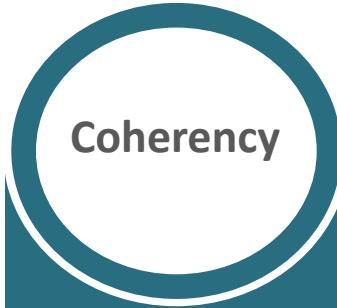
$$PMI(t_1, t_2) = \log \frac{Pr(t_1, t_2|D)}{Pr(t_1|D)Pr(t_2|D)},$$

$Pr(t_1, t_2|D)$  : the probability of the two terms to co-occur in the corpus.

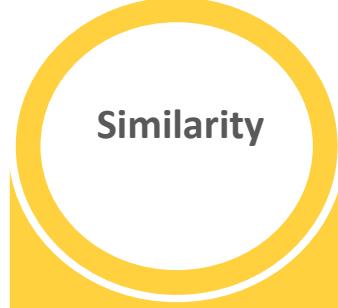
# Statistical approaches



Co-occurrence  
of  
query terms



Inter-similarity  
between  
documents  
containing query  
term



Queries that are  
more similar to  
collection are  
easier to satisfy



More specific  
queries are easier  
to satisfy

# Frequency-based Specificity Metrics

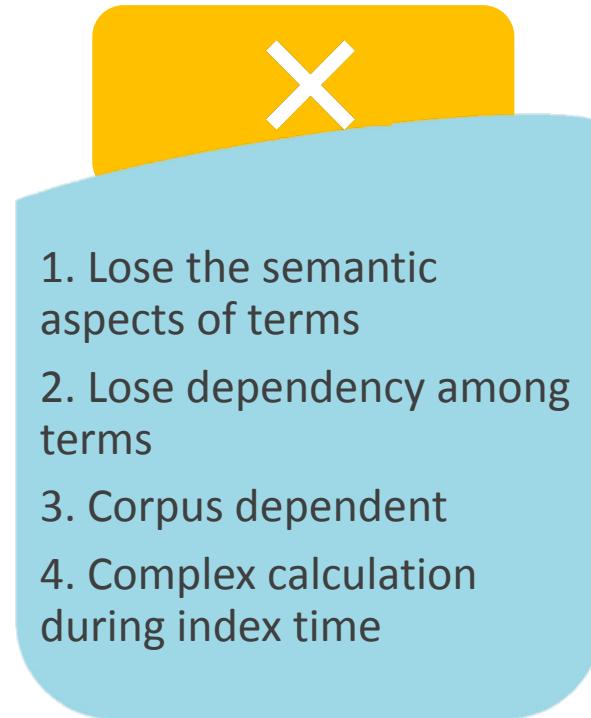


1. Preserve statistical features of terms



1. Lose the semantic aspects of terms
2. Lose dependency among terms
3. Corpus dependent
4. Complex calculation during index time

# Frequency-based Specificity Metrics

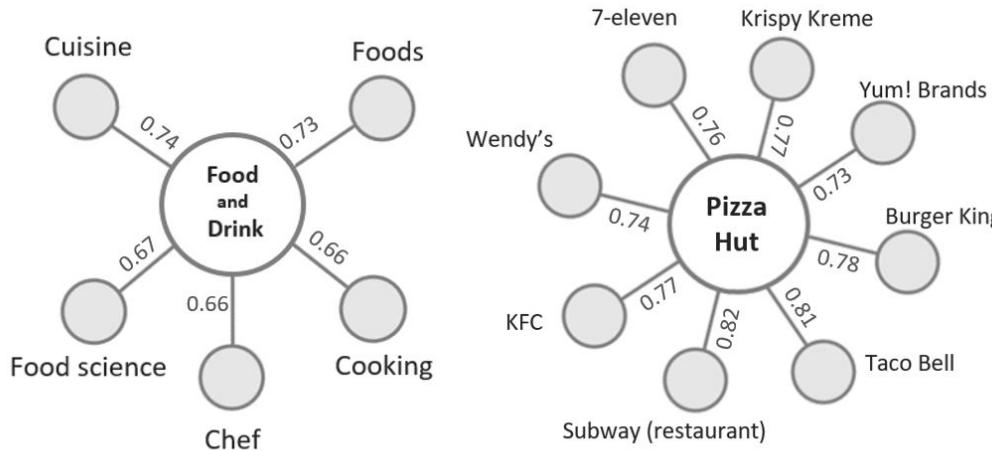


# Neural-embedding based QPP

**$\varepsilon$ -neighborhood:** Selected local neighborhood surrounding an embedding vector of term  $t_i$ , by retrieving a set of highly similar terms to  $t_i$ .

$$N_\varepsilon(t_i) = \{t_j : \frac{v_{t_i} \cdot v_{t_j}}{\|v_{t_i}\| \|v_{t_j}\|} \geq \varepsilon \times \mu(t_i)\}$$

*Degree of similarity of most similar term to  $t_i$ .*

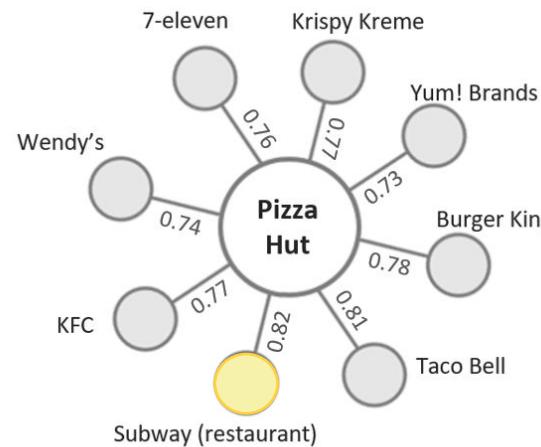
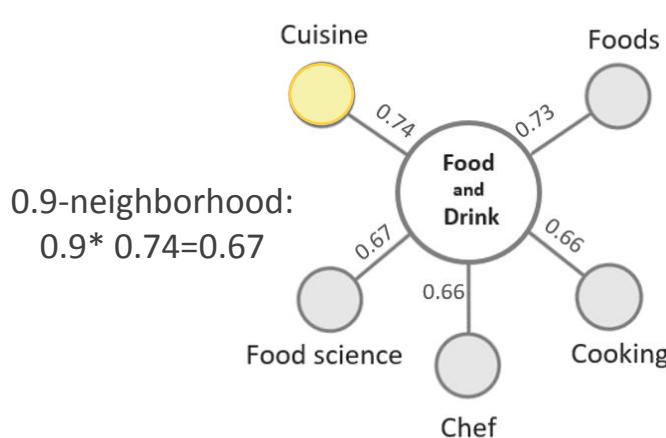


# Neural-embedding based QPP

**$\varepsilon$ -neighborhood:** Selected local neighborhood surrounding an embedding vector of term  $t_i$ , by retrieving a set of highly similar terms to  $t_i$ .

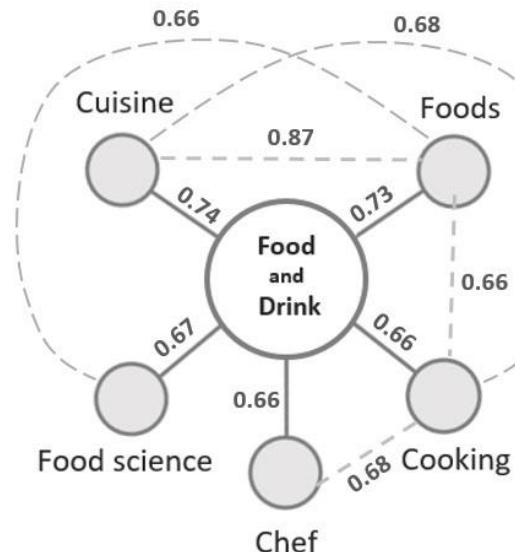
$$N_{\varepsilon}(t_i) = \{t_j : \frac{v_{t_i} \cdot v_{t_j}}{\|v_{t_i}\| \|v_{t_j}\|} \geq \varepsilon \times \mu(t_i)\}$$

*Degree of similarity of most similar term to  $t_i$ .*

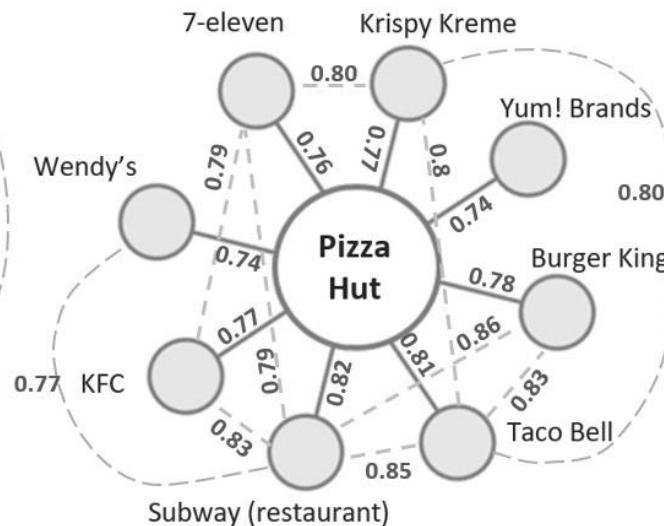


# Neural-embedding based QPP

**Ego network:**  $t_i$  is the ego node and is connected directly to other terms only if the degree of similarity between the ego and its neighbors is above a given threshold.



(a)

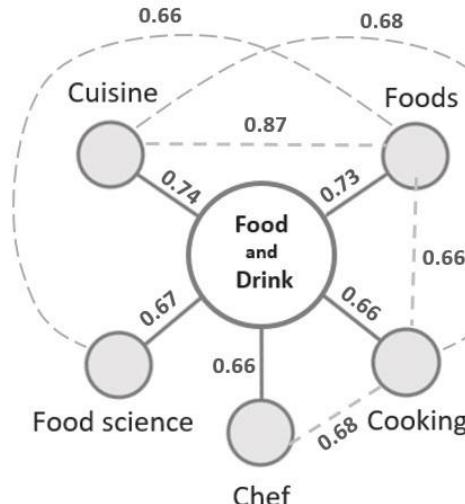


(b)

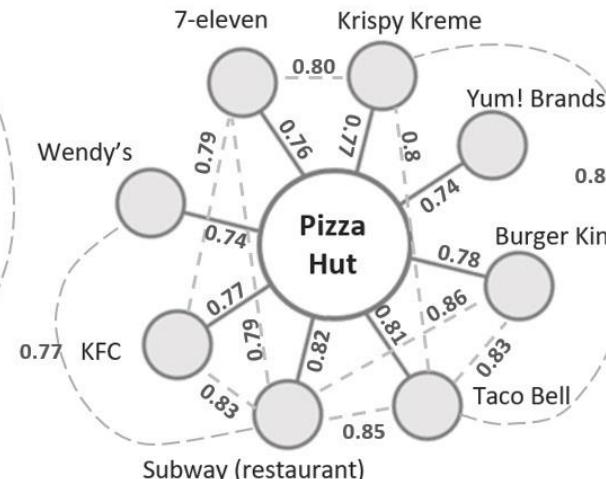
# Neural-embedding based QPP

## Intuition

- ✓ A specific term is likely to be associated with a large number of specific terms in its neighborhood.
- ✓ Highly specific terms → precise semantics → likelihood of being surrounded by a higher number of specific terms



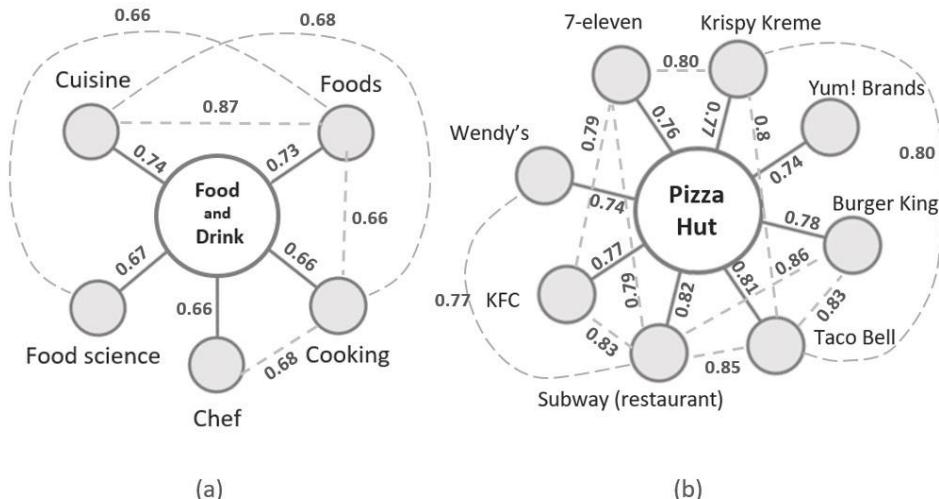
(a)



(b)

# Neural-embedding based QPP

- Neighborhood size (NS)
- Weighted Degree centrality (WDC)
- Median Absolute Deviation (MAD)
- Neighborhood Variance (NV)
- Most Similar Neighbor (MSN)
- Neighborhood Vector similarity (NVS)



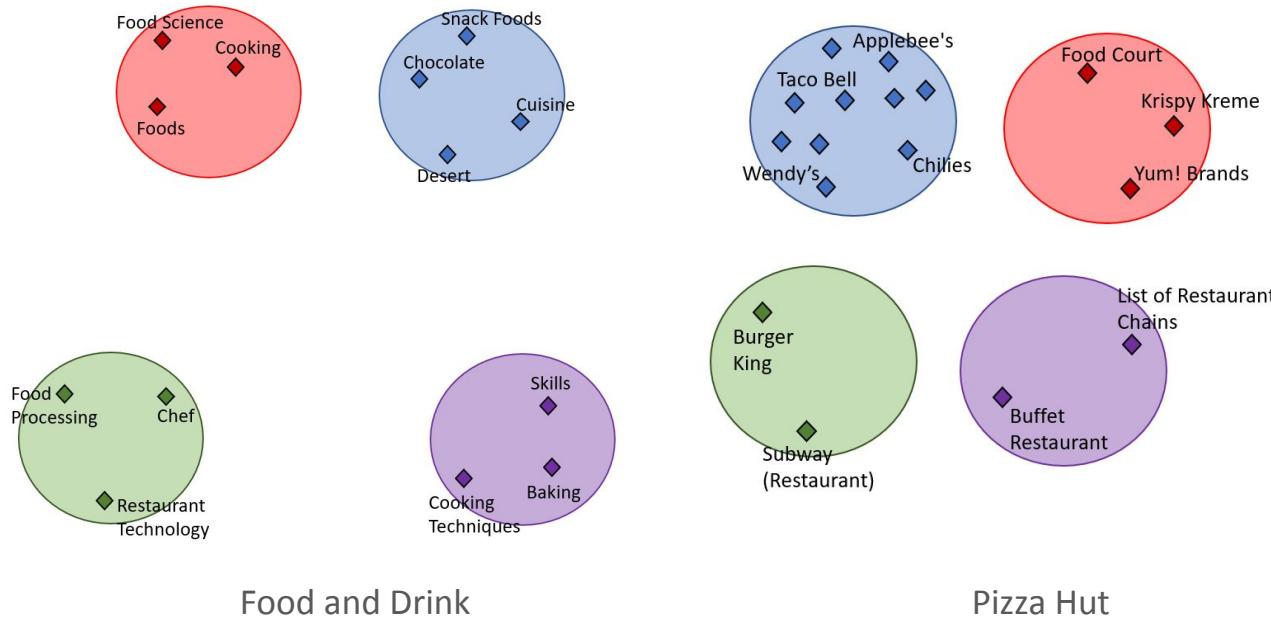
(a)

(b)

# Neural-embedding based QPP

## Centroid Network steps:

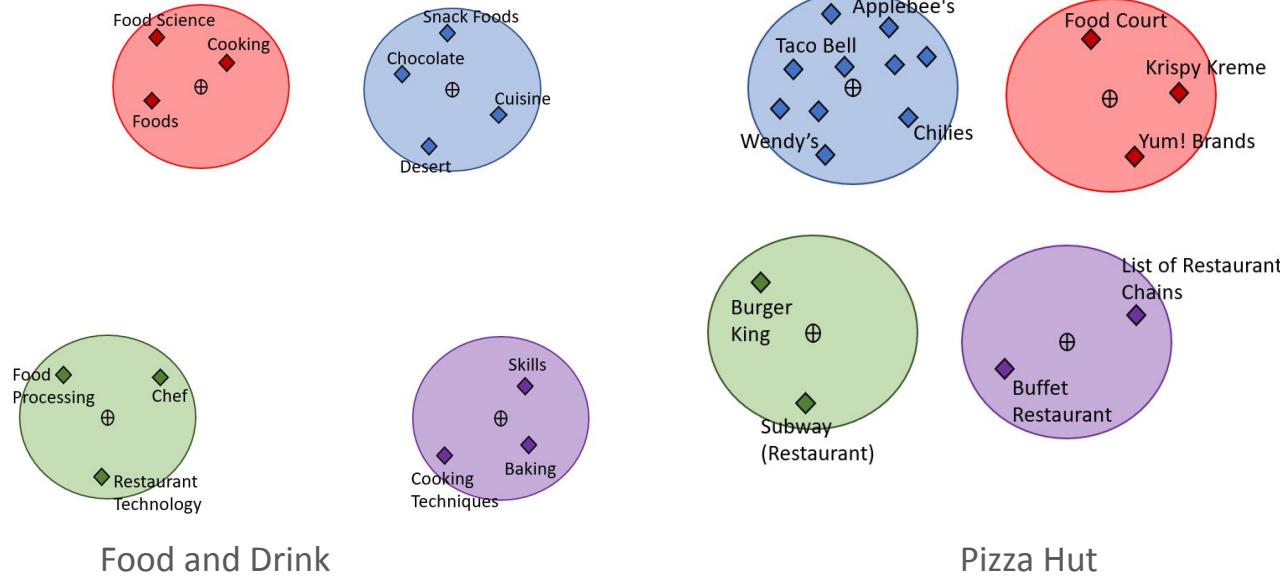
- 01 We applied K-means clustering algorithm to find K cluster for term  $t_i$   $C_{t_i}^1, \dots, C_{t_i}^K$



# Neural-embedding based QPP

## Centroid Network steps:

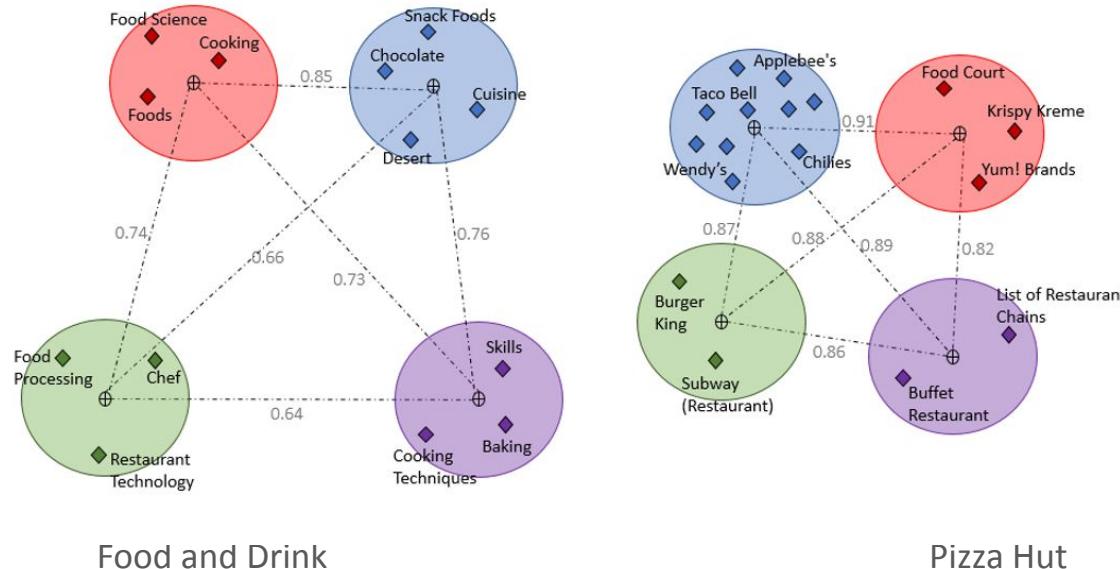
- 01 We applied K-means clustering algorithm to find K cluster for term  $t_i$   $C_{t_i}^1, \dots, C_{t_i}^K$
- 02 We find the centroid of each cluster



# Neural-embedding based QPP

## Centroid Network steps:

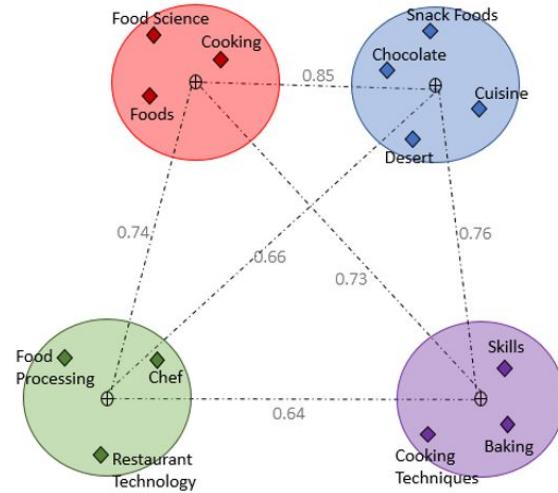
- 01 We applied K-means clustering algorithm to find K cluster for term  $t_i$   $C_{t_i}^1, \dots, C_{t_i}^K$
- 02 We find the centroid of each cluster
- 03 We make a weighted graph by connecting all the centroid



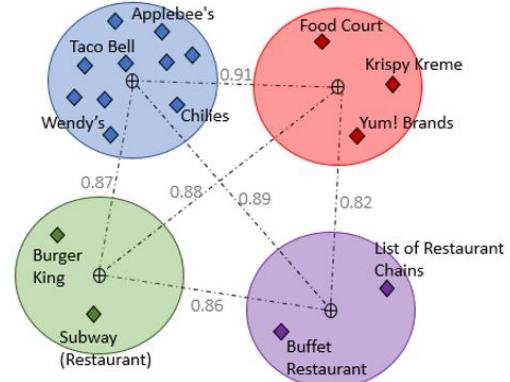
# Neural-embedding based QPP

## On Centroid Network :

- Edge Weight Avg\_centeroid (EWAc)
- Edge Weight Max\_centeroid (EWXc)
- Cluster Elements Variance (CEV)



Food and Drink



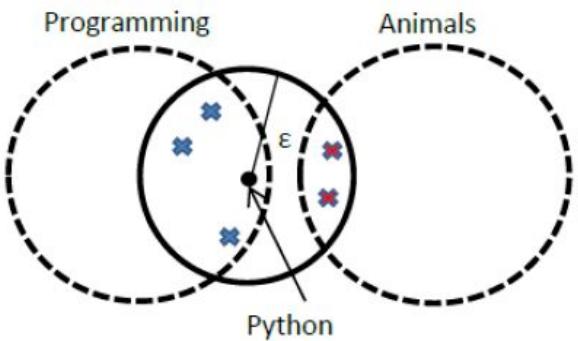
Pizza Hut

# Neural-embedding based QPP - Clarity

**Idea:** Using different senses of the query as an indicator of query ambiguity

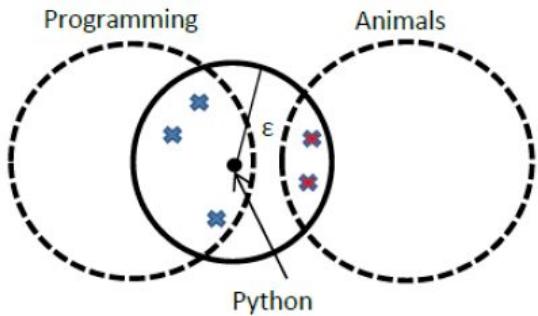
Ambiguous queries

- Example: "python" or “python”



# Neural-embedding based QPP - Clarity

**Idea:** Using different senses of query as an indicator of query ambiguity



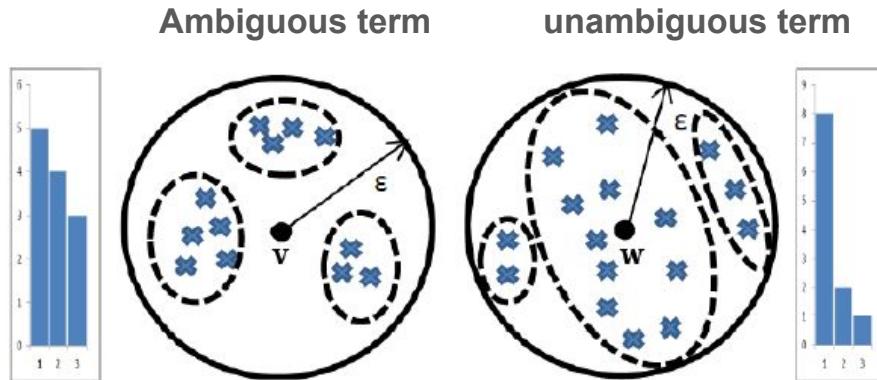
The ambiguity of a term is determined exclusively by its occurrences within the target corpus.

- 'python' could be unambiguous if the target collection consisted only of zoological reports.
- Gaussian Mixture Model (GMM): Estimates query term ambiguity by analyzing the local neighborhood in embedded space of word vectors.

# Neural-embedding based QPP

E-neighbourhood of query terms:

$$N_\epsilon(\mathbf{q}) = \{\mathbf{x} : 0 \leq \cos^{-1} \left( \frac{\mathbf{x} \cdot \mathbf{q}}{|\mathbf{x}| |\mathbf{q}|} \right) < \epsilon\}$$



Illustrative diagram of the neighborhood of an ambiguous word with multiple senses

- Gaussian Mixture Model (GMM) of K components.
- Each Gaussian component in the neighbourhood of a query term potentially corresponds to a sense of the query term.
- The variance of the prior values is high

# Contextualized-embedding based QPP

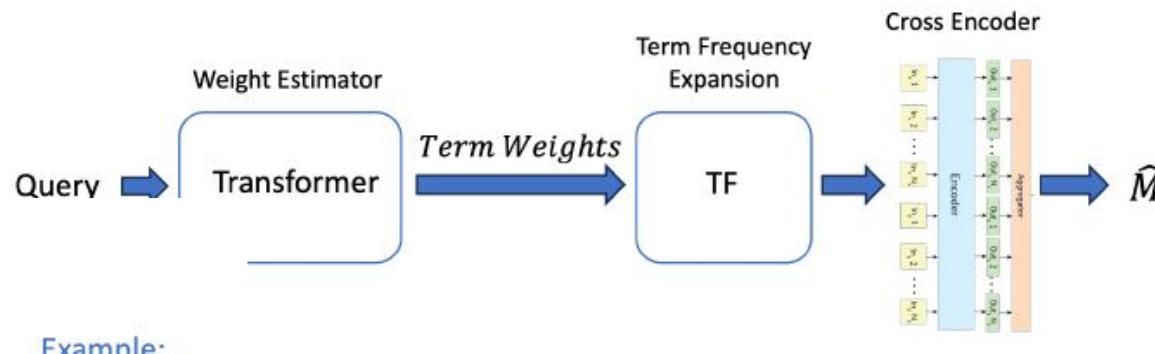
**Idea:** Learning the performance from different query variants

- **Identifying Term Impact:** Determining which query terms impact query performance positively or negatively.
- **Learning Query Term Weights:** learning weights for query terms to understand their positive or negative contribution to performance.
  - **Easy Queries:** Queries with terms contributing positively are likely to be easier
  - **Hard Queries:** Queries with many terms with a negative impact are considered harder

# Contextualized-embedding based QPP

## Approach:

1. Developing pairs of queries addressing the same information need but with different retrieval effectiveness.
2. Learning the likelihood of query terms contributing to the query's softness or hardness.
3. Adopting learned term likelihoods to estimate query performance.



# Contextualized-embedding based QPP

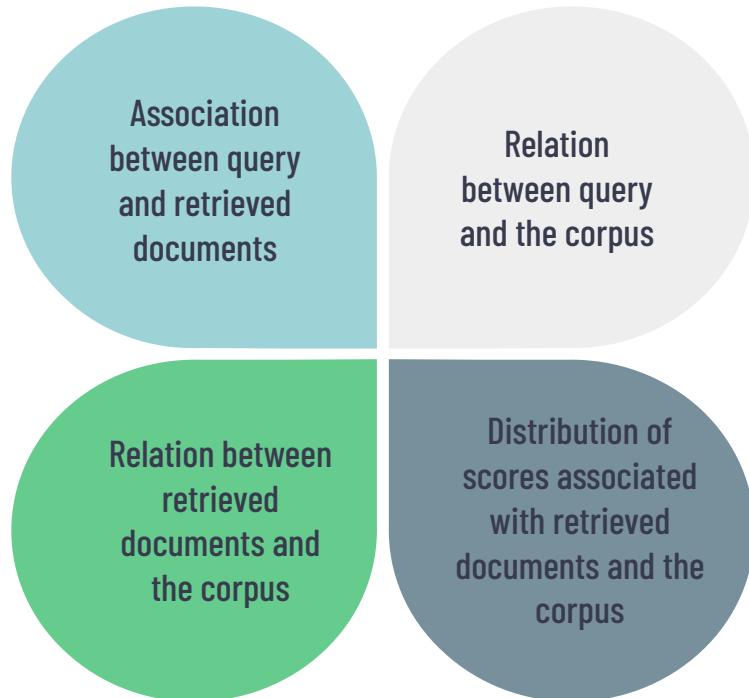
## Contextual Adaptation of Term Difficulty:

product level activity define	define a multichannel radio
definition of capias issued on a background	how far back do employment background checks
what is the gas called that they give you at the dentist	calculate the mass in grams of 2.74 l of co gas

# Q & A

# Post-retrieval QPP

# Post retrieval QPP



# Coherency-based QPP

**Idea:** “Coherency” of the result-list with respect to the corpus.

- The extent to which top results use the same language.

**Intuition:**

- A common language of the retrieved documents.
- Being distinct from general language of the whole corpus is an indication of high quality.

**Discrepancy between:**

- Likelihood of words most frequently used in retrieved documents
- Likelihood in the whole corpus.

# Coherency-based QPP - Clarity

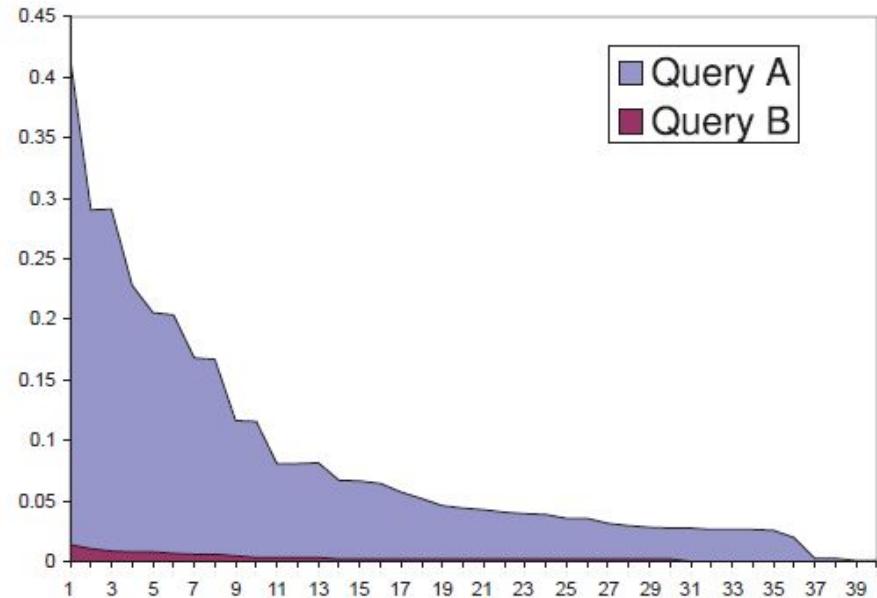
**Clarity:** KL-divergence between the language model of the result set and the language model of the entire collection.

$$Clarity(q) = KL_{div}(Pr(\cdot|D_q) || Pr(\cdot|D)) = \sum_{t \in V(D)} Pr(t|D_q) \log \frac{Pr(t|D_q)}{Pr(t|D)}$$

- **Potential downside:** efficiency
- **Solution:**
  - Precompute the collection's language model at indexing time.
  - Sum over all documents in the result set.

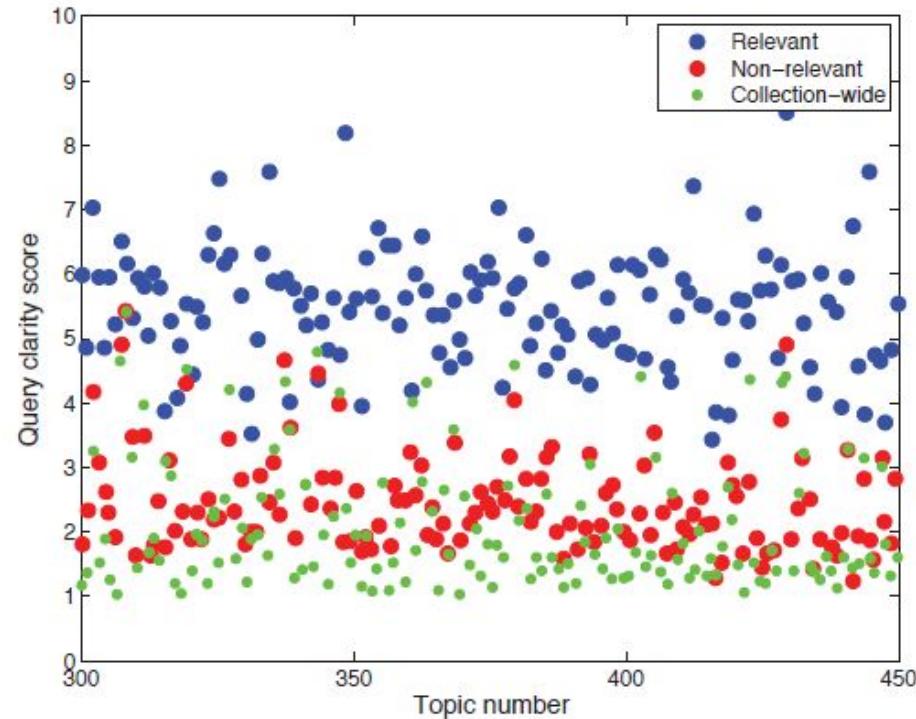
# Coherency-based QPP

- **Query A:** “What adjustments should be made once federal action occurs?”
- **Query B:** “Show me any predictions for changes in the prime lending rate and any changes made in the prime lending rates”



- **Clarity score:** area under the graph.

# Coherency-based QPP



# Coherency-based QPP

So far:

- The associations between the **query and the retrieved documents**.
- The relation between the **corpus and the retrieved documents**.

How about the **association among the retrieved set of documents themselves?**

- Coherency of the retrieved set of documents can be an indication of query difficulty.
- Motivated by the Cluster hypothesis.

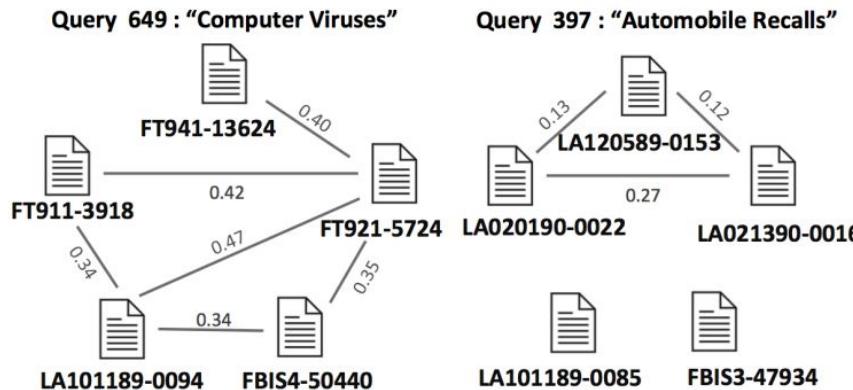
**Assumption:**

- Coherent set of retrieved documents.
- The retrieval method can discriminate between relevant and non-relevant documents.

# Coherency-based QPP

## Building the Network

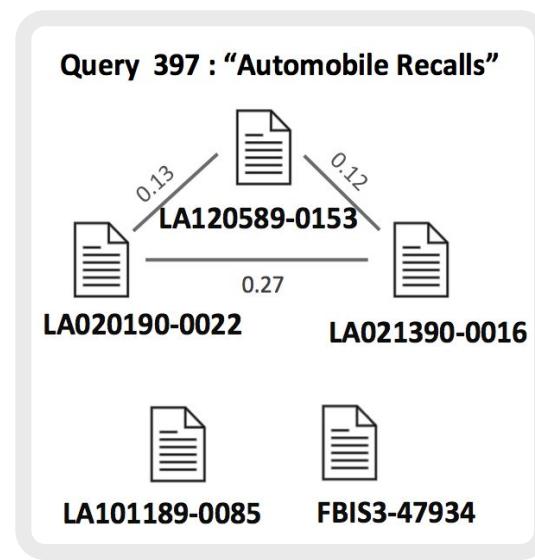
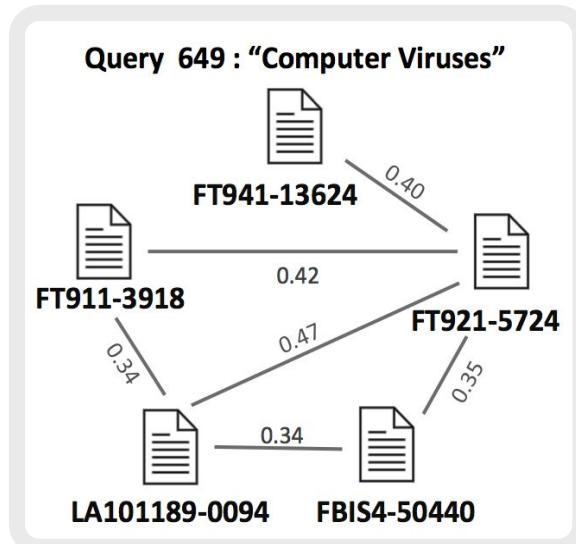
- A host of coherence measures based on **the graphical modeling of the retrieved documents**.
- Building a **weighted undirected document association network** that captures the retrieved documents and their similarities.
- **Query coherence** as a function of the characteristics of the document association network.



# Coherency-based QPP

- **Document Association Network:**
  - Fully connected graph that finds.
  - All pairwise document similarities.
  - Top-k documents retrieved for query q.
  
- **Pruning:**
  - Sparser network
  - Remove nodes with negligible weights.
  - Remove edges below the average weight.

# Coherency-based QPP



04402-PSIBA  
P600-68JL0JAI  
PE0.0  
PEP4-ESIBA

2800-68JL0JAI  
PEP4-ESIBA  
PE0.0

# Coherency-based QPP

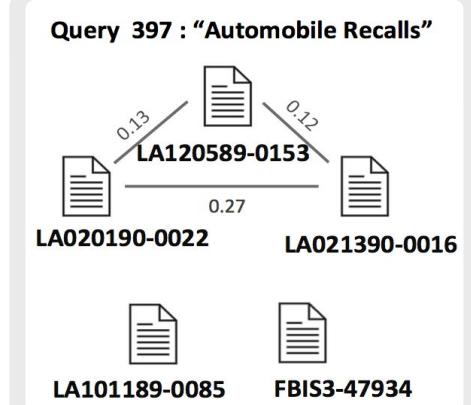
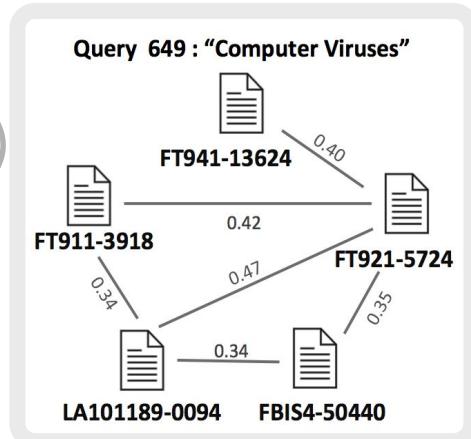
Higher number  
of document  
associations

Denser

Lower edge  
weights

Disconnected

Sparse



# Coherency-based QPP

Average Clustering coefficient (ACC)

Average Degree Connectivity (ADC)

Average Neighbourhood Degree (AND)

Density (D)

Higher number of document associations

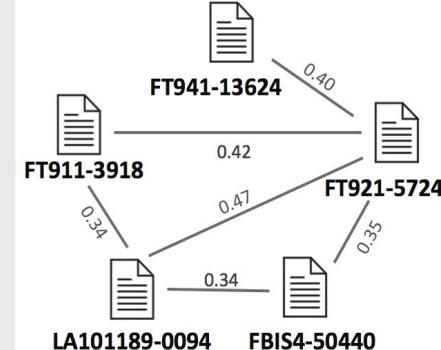
Denser

Lower edge weights

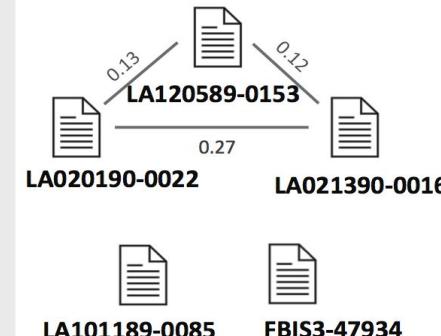
Disconnected

Sparse

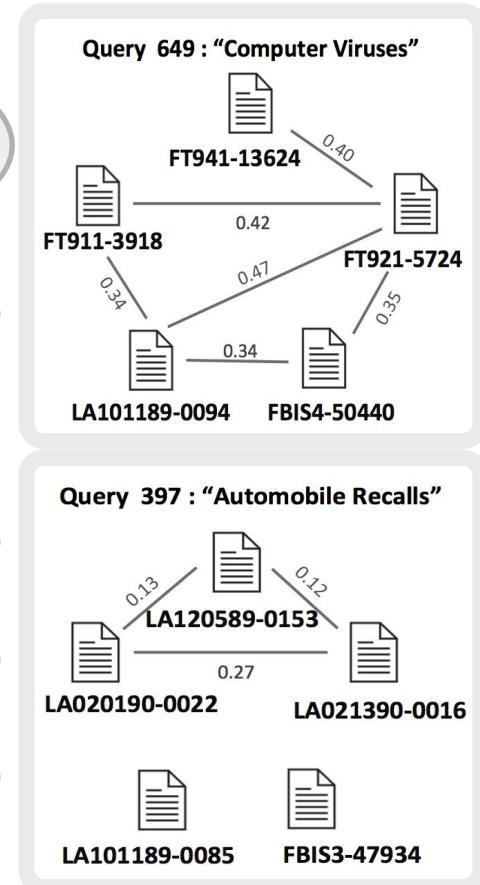
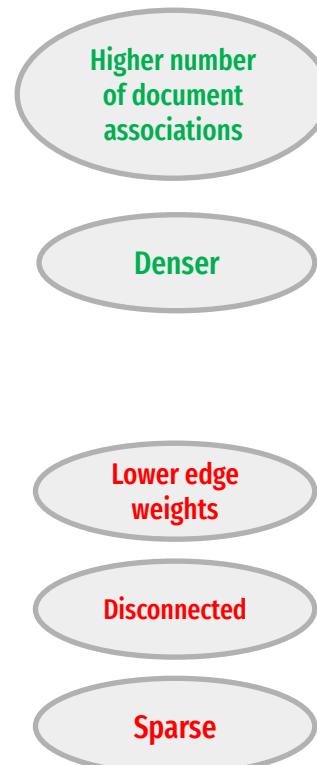
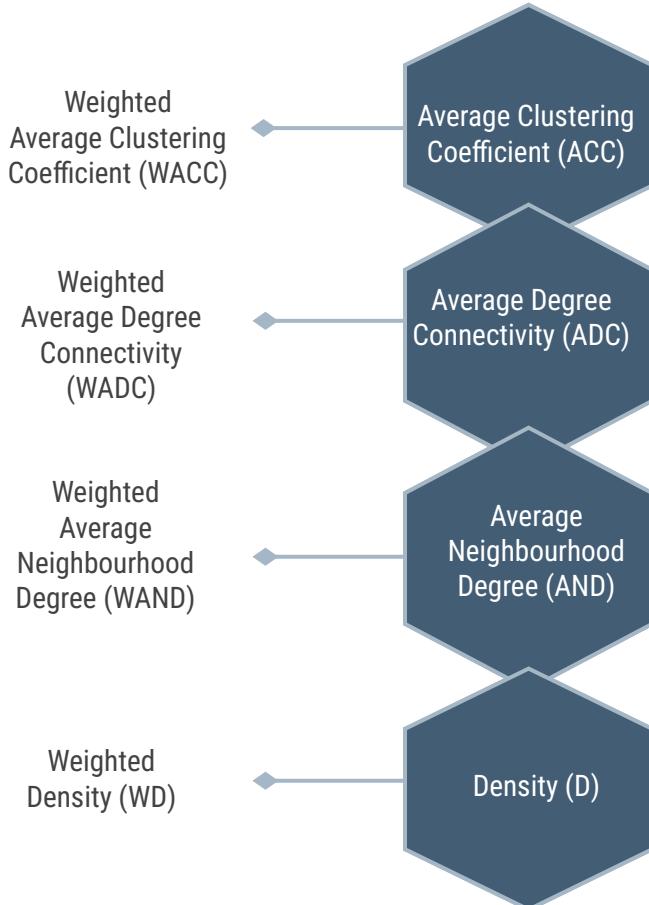
Query 649 : "Computer Viruses"



Query 397 : "Automobile Recalls"



# Coherency-based QPP



# Robustness-based QPP

# Query perturbation

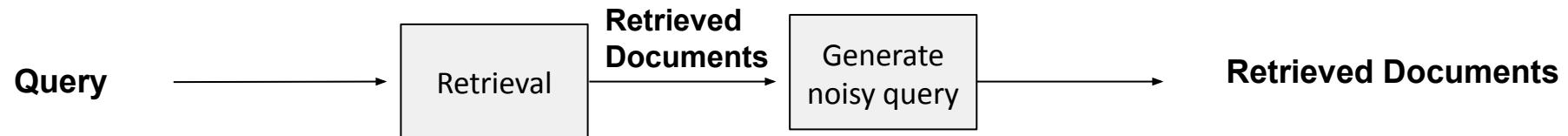
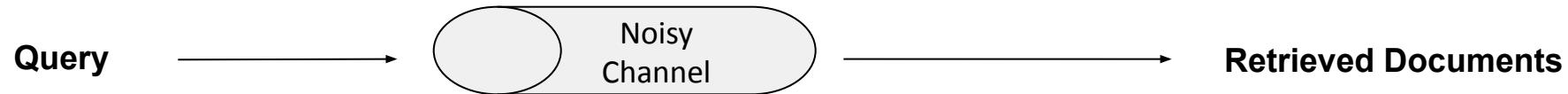
## Idea:

- Small modifications to the query.
- Robustness of the results list.

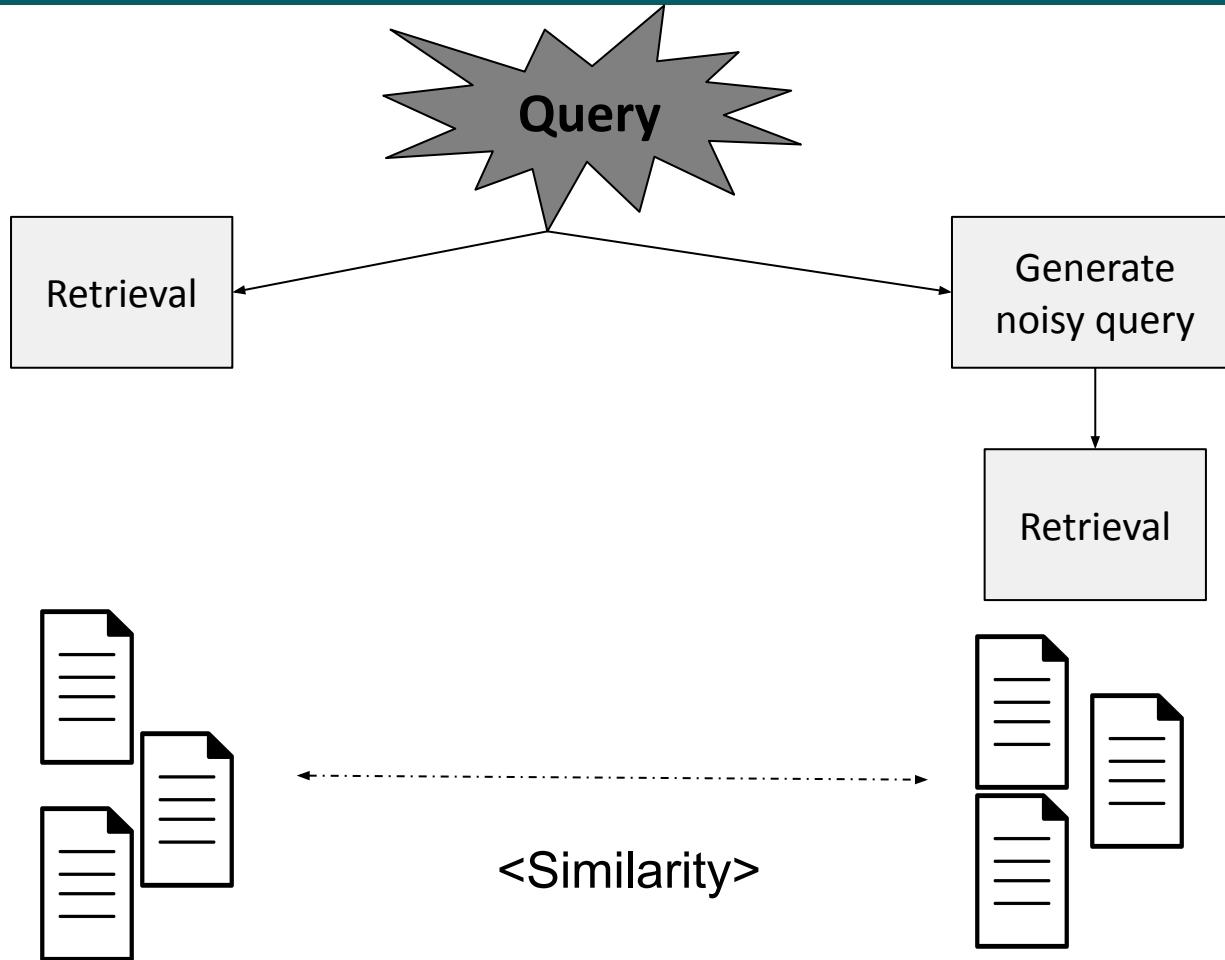


# Query perturbation

**Query Feedback:** Models retrieval as a communication channel problem.



# Query perturbation



# Query perturbation

**Idea:** Perturbation with sub-queries.

**Approach:**

- Query.
- Sub-queries of individual terms.
- Overlap between the results lists.

**Interpretation:**

- A difficult query would be one where the query is not dominated by a single keyword.

# Query perturbation

**Idea:** Injecting noise in the semantic space to the vector representation of the query.



Dense retrievers encode queries and documents within a low-dimensional embedding space.



Generate query perturbations for measuring query robustness



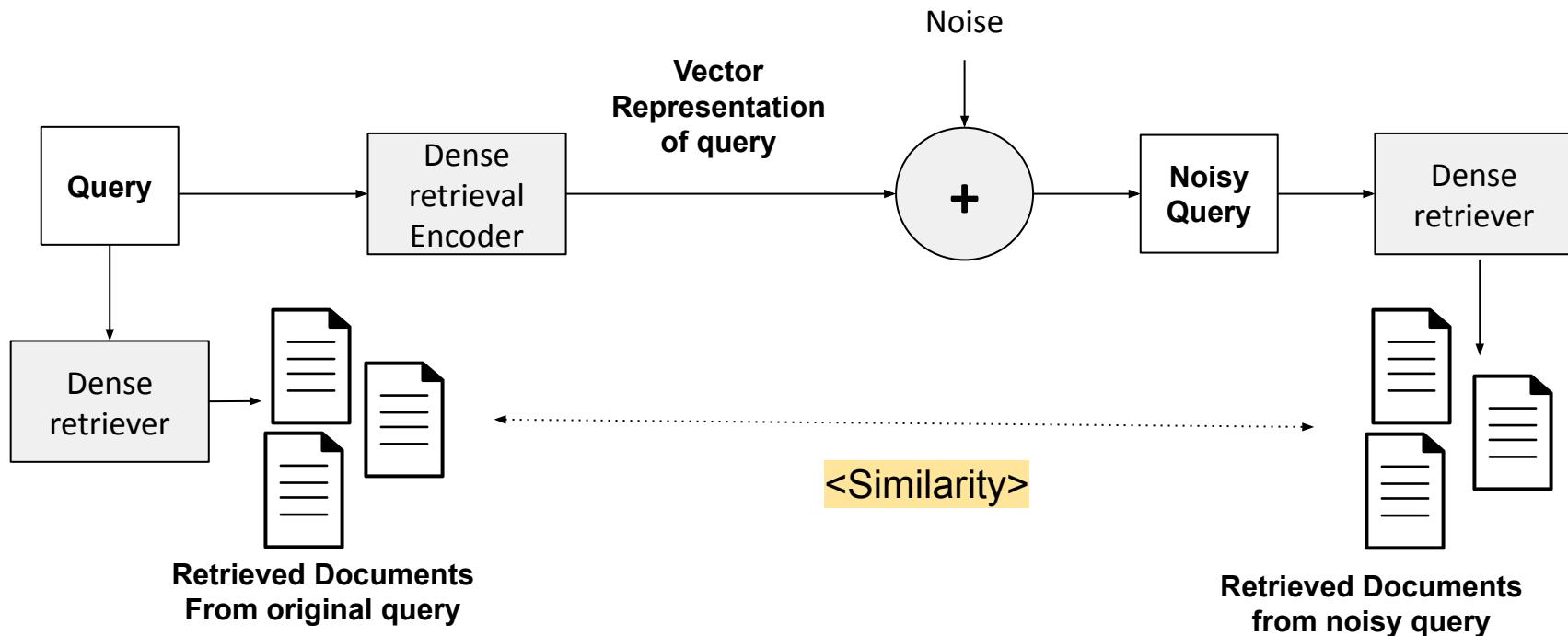
Systematically injecting noise into the contextualized neural representation of each query



A less robust query would be one that would experience a noticeable change in its retrieval.

# Query perturbation - for dense retrievers

**Idea:** Injecting noise in the semantic space to the vector representation of the query.



# Retrieval perturbation

**Idea:** Query robustness with respect to using different retrieval methods.

**Approach:**

- Retrieve results using ranker A.
- Retrieve results using ranker B.
- High overlap in results retrieved by A and B.
  - High agreement on the set of relevant results.
  
- Submitting the query to different retrieval methods and measuring the diversity of the ranked lists obtained.

# Score-based QPP

# Score-based QPP

Drawbacks of clarity or the robustness based approaches – time consuming

Alternative: analyzing the score distribution of the result set to identify query difficulty.

Retrieval score: Reflecting similarity of documents to queries

→ The distribution of retrieval scores can potentially help predict query performance.

Increase in retrieval-score → more relevant results

The difference between retrieval scores → “discriminative power” of the query.

# Score-based QPP - WIG

**Idea:** Measuring the divergence between the mean retrieval score of top-ranked documents and that of the entire corpus.

**Hypothesis:** the more similar these documents are to the query, with respect to the query similarity exhibited by a general non-relevant document (i.e., the corpus), the more effective the retrieval.

$$WIG(q) = \frac{1}{k} \sum_{d \in D_q^k} \sum_{t \in q} \lambda(t) \log \frac{Pr(t|d)}{Pr(t|D)}$$

$\lambda(t)$  : normalization w.r.t query length.

# Score-based QPP - NQC

**Idea:** measuring how distinguishable the retrieved results are

Can we easily distinguish the relevant and irrelevant stuff?

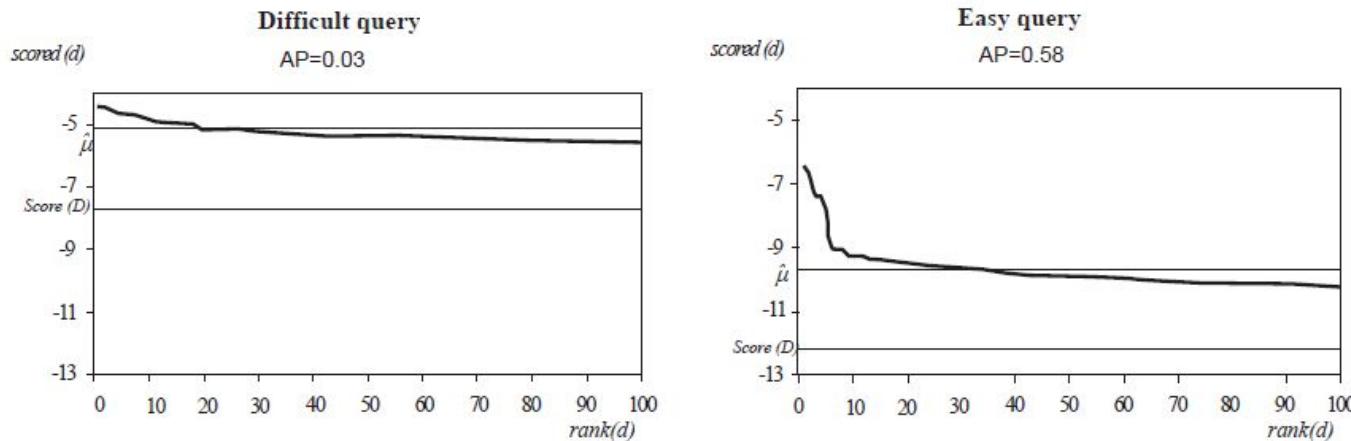
Higher variance in scores → easier distinguishability of items

$$NQC(q) = \frac{\sqrt{\frac{1}{k} \sum_{d \in D_q^k} (Score(d) - \mu)^2}}{|Score(D)|}$$

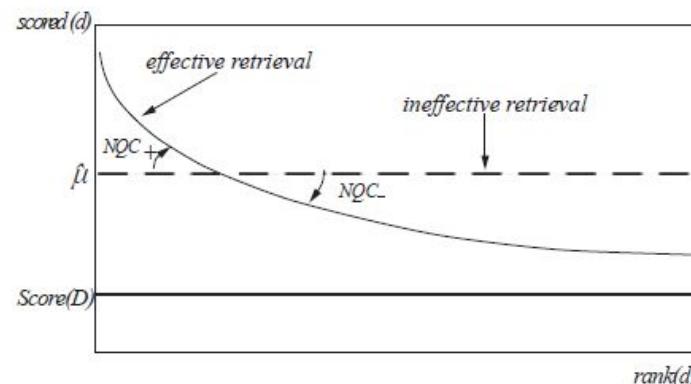
Measuring standard deviation  
of retrieval scores in the  
top-retrieved document

normalizing if by the whole  
collection score

# Score-based QPP - NQC



**General interpretation**

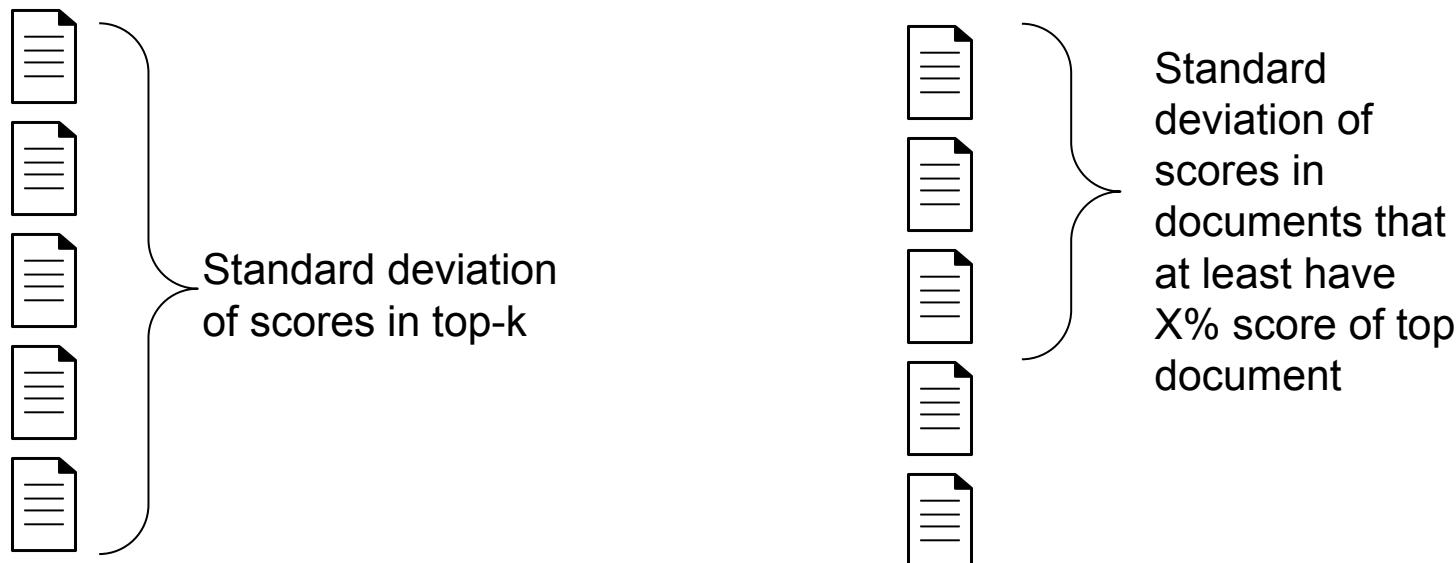


# Score-based QPP - Dynamic Cut off

**Idea:** Choose top-K retrieved document dynamically

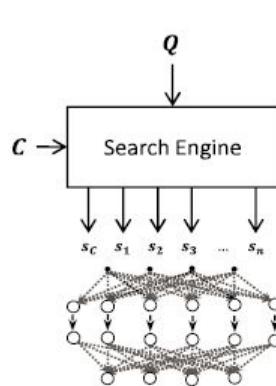
Instead of constant depth → keep documents with a score greater than a certain percentage ( $x$ ) of the top score.

For example, if we choose  $x = 90\%$ , all documents that have a score of at least 90% of the top score are included in the standard deviation calculation.

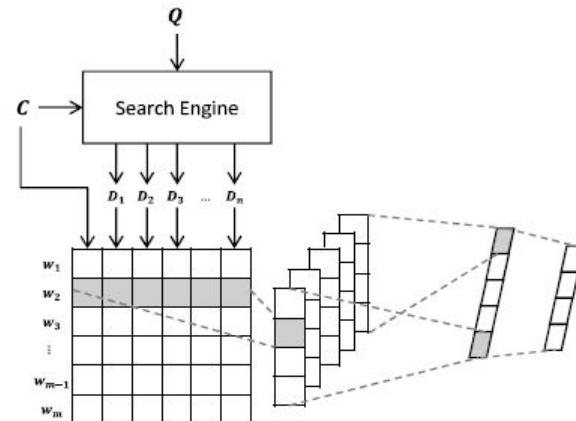


# Embedding-based QPP Post retrieval

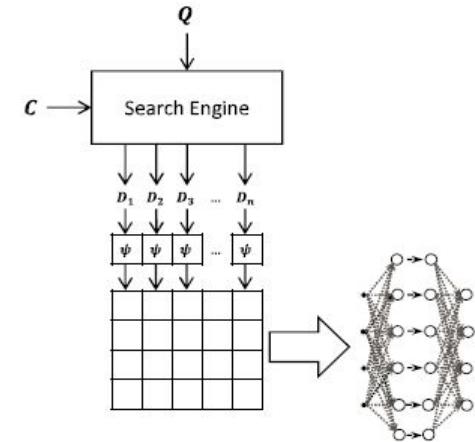
# Neural-based QPP



(a) Retrieval Scores Analyzer



(b) Term Distribution Analyzer

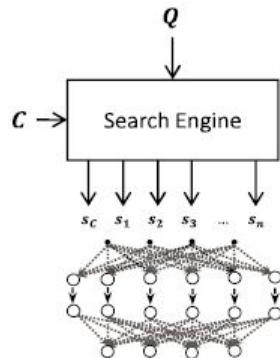


(c) Semantic Analyzer

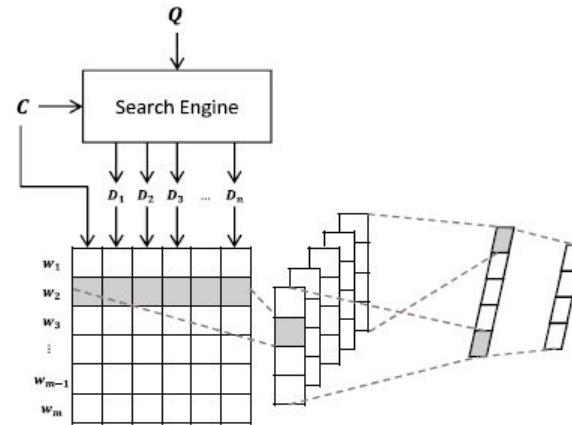
$$s_i = \begin{cases} \text{score}(q, C) & \text{if } i = 1 \\ \text{score}(q, D_{i-1}) & \text{o.w.} \end{cases}$$

**Idea:** Learning different representation → Aggregating them using the arithmetic mean and then fed into a fully-connected feed-forward network to produce a single score for query performance prediction

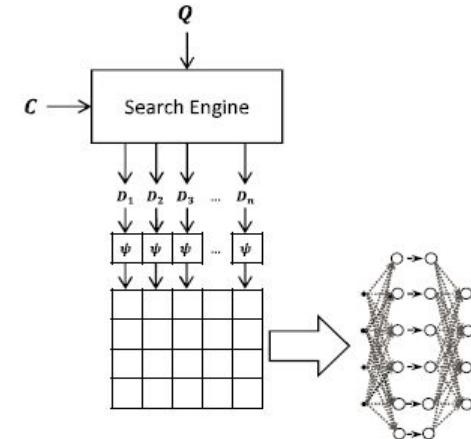
# Neural-based QPP



(a) Retrieval Scores Analyzer



(b) Term Distribution Analyzer



(c) Semantic Analyzer

**Approach:** Training for optimizing across other QPP models as weak labels.  
Simultaneously optimizes  $N$  loss functions, each corresponding to a weak label.

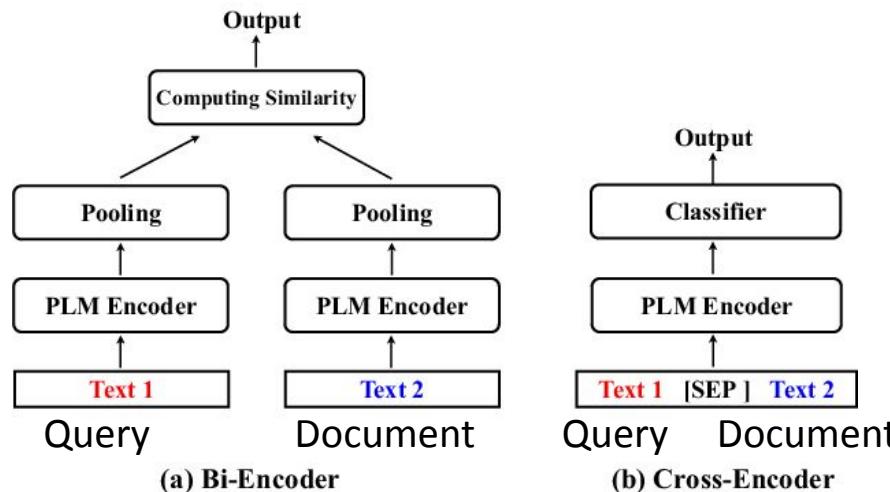
Point wise and pairwise style.

**Drawback:** Lots of noise in QPP signals - requires lots of data

# BERT-QPP

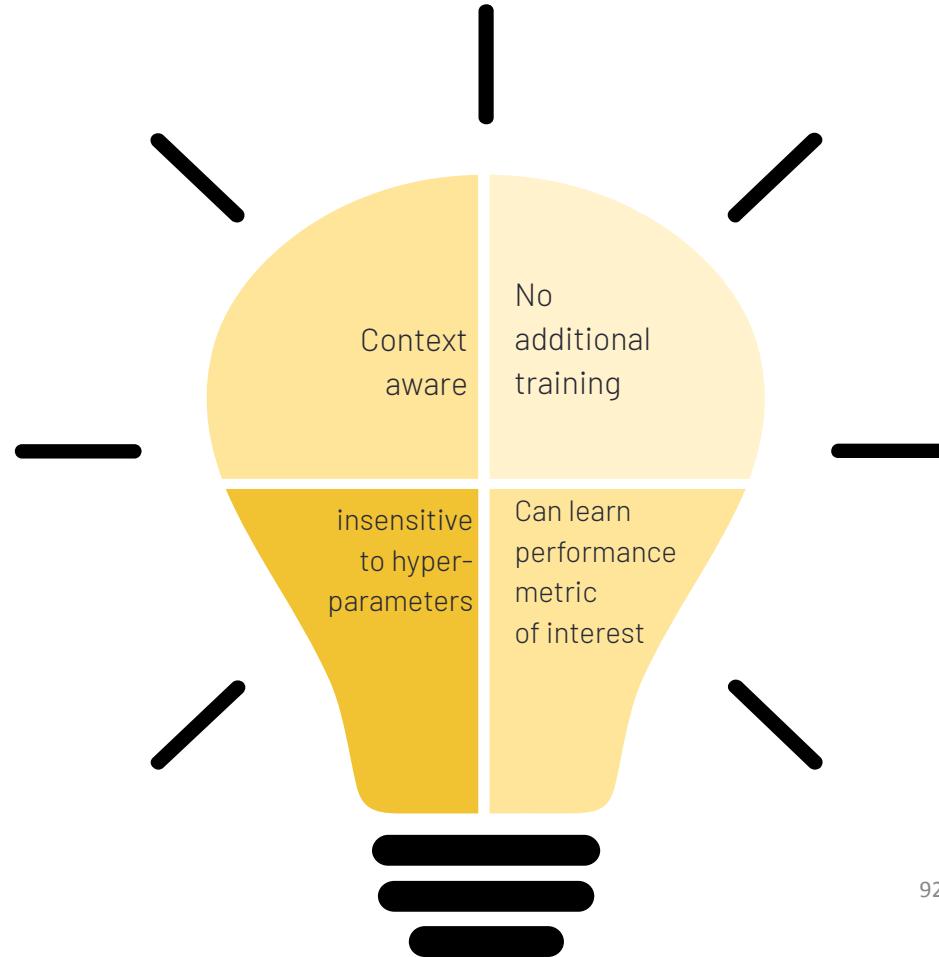
**Idea:** Directly learns query performance through the fine-tuning of BERT

- learning a continuous difficulty score based on the association between the input query and the top- $k$  retrieved documents in response to  $q$
- Learning the relevance → Learning the performance



# BERT-QPP

- Two widely adopted architecture
  - Cross-encoder → BERT-QPPcross
  - Bi-encoder → BERT-QPPbi



# BERT-QPP

	BERT-QPP <sub>bi</sub>	Bert-QPP <sub>cross</sub>
Number of Interactions		
Capturing association between query and document space		
Offline Computation		
Inference Time		

# BERT-QPP

- Comparing the inference time of neural-based QPP baselines when run on an RTX3090 GPU.
- Bi-encoder architecture shows significantly lower inference time ( $4 \times$  smaller) compared to the cross-encoder network.
- Query latency for BM25 “ 55ms per query”
- Delay caused by BERT-QPP methods can be tolerable.

Method	Inference time per query (ms)
NQA-QPP	25.3
NeuralQPP	21.3
BERT-QPP <sub>cross</sub>	2.6
BERT-QPP <sub>bi</sub>	0.7

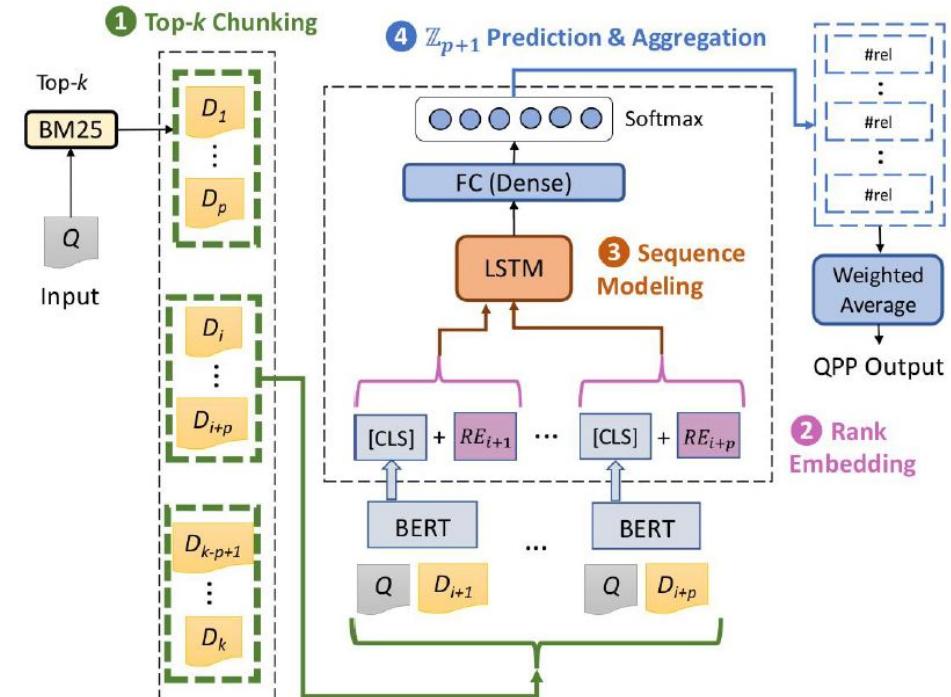
# qppBERT-pl

**Idea:** addressing limitations of top-retrieved documents

- Considering position
- Considering all the top-k retrieved documents

**Approach:**

- partitioned top-k documents into  $\lfloor k/p \rfloor$  chunks, each of size  $p$ .
- The query-document cross-encoded representations + positional embeddings fed into LSTMs



# Enriched BERT-QPP

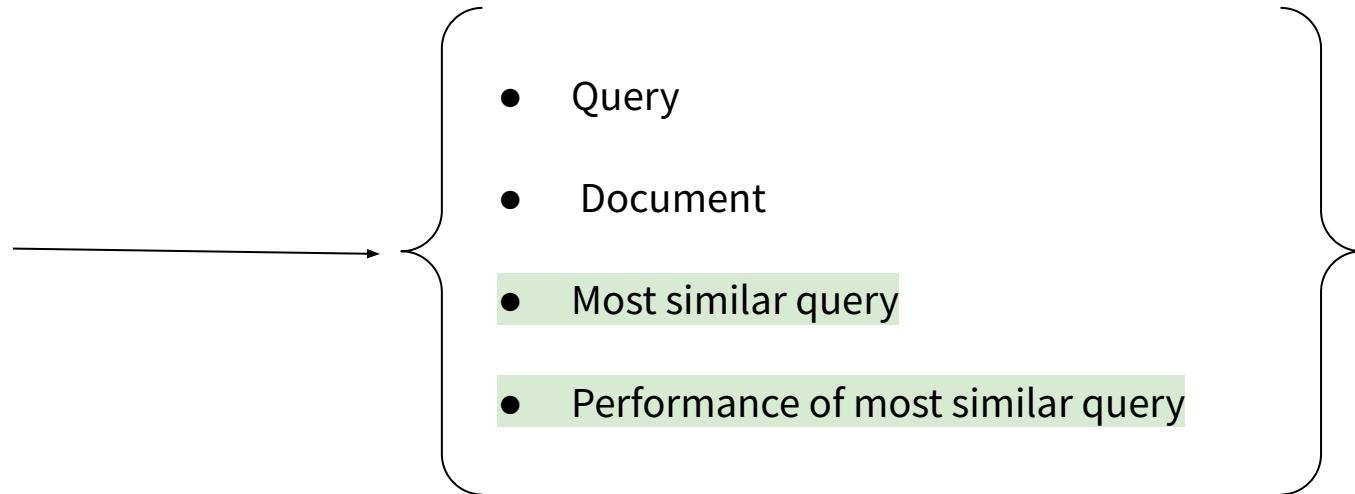
**Idea:** leveraging from the performance of known query.

**Assumption:** Having a query store with known performance

**Approach:** Injecting the performance of known queries as the input text to BERT-QPP

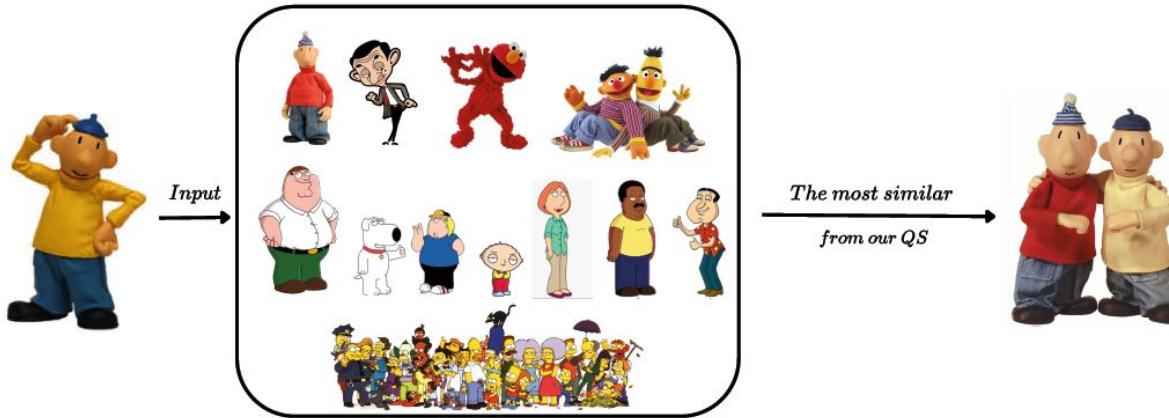
BERT-QPP Inputs:

- Query
- document



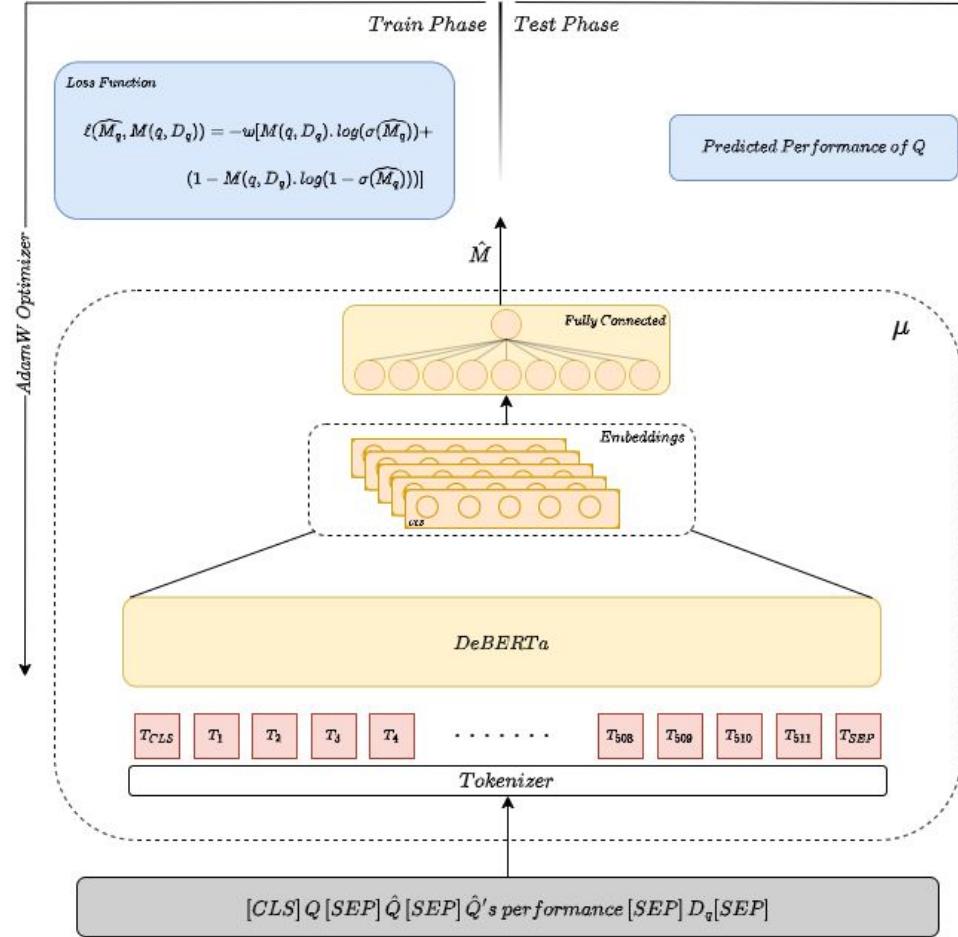
# Enriched BERT-QPP

Finding Nearest Neighbor queries for a given query:



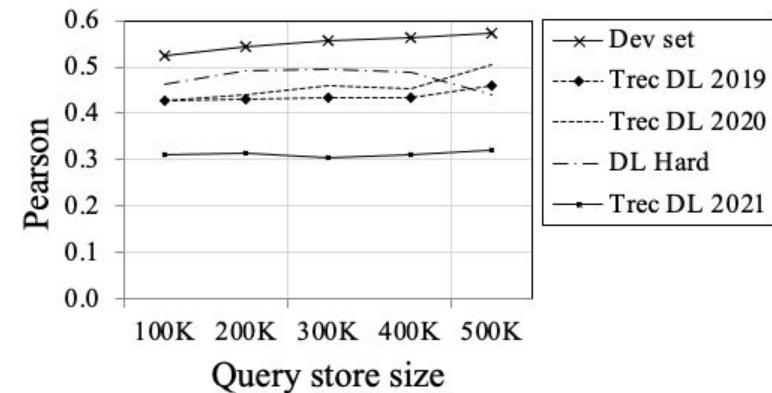
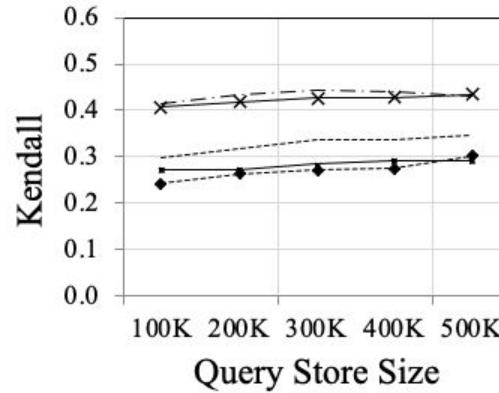
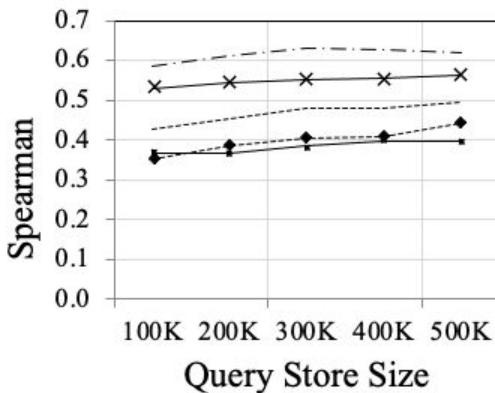
Query		Most similar query from QS	
qid	text	qid	text
190044	foods to detox liver naturally	189691	foods that naturally detox the liver
2	Androgen receptor define	914258	what type of receptor is androgen
786674	what is prime rate in canada	481686	prime rate canada definition
1048876	who plays young dr mallard on ncis	1048416	who plays on ncis tv show
1110199	what is wifi vs bluetooth	404536	is bluetooth wifi
489204	right pelvic pain causes	583919	what cause pelvic pain

# Enriched BERT-QPP



# Enriched BERT-QPP

Impact of size of query store :



→ Fairly robust w.r.t query store size

# Learning to Rank and Predict

## Objective:

- learning to perform ad hoc retrieval while at the same time learning to predict the quality of the performance of a query through a **multi-task learning framework**.

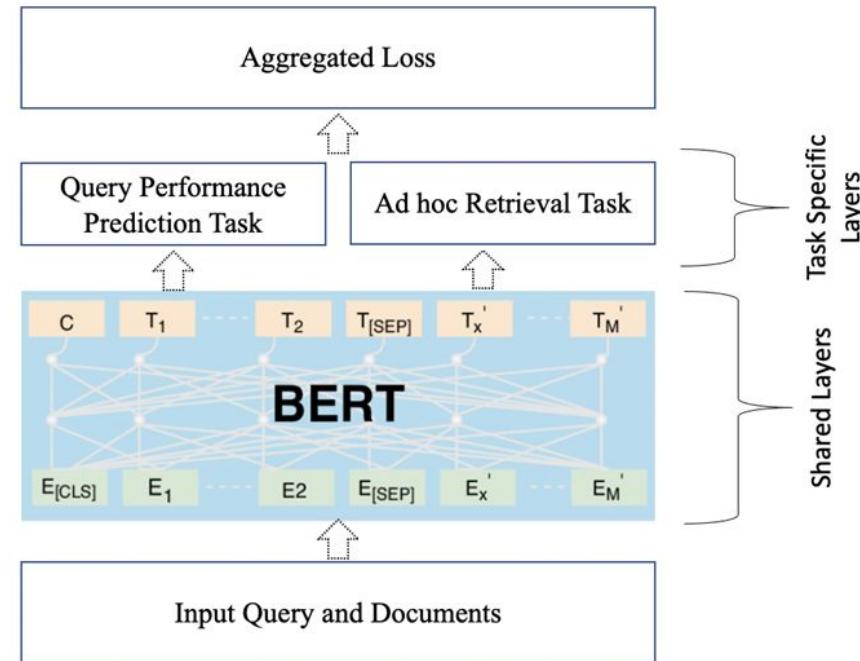
## Hypothesis:

- Learning to rank and learning to predict query performance simultaneously will result in more effective ranking and more accurate performance prediction given the synergies between the two tasks.

# Learning to Rank and Predict

## Multi-task Query Performance Prediction Framework (M-QPPF)

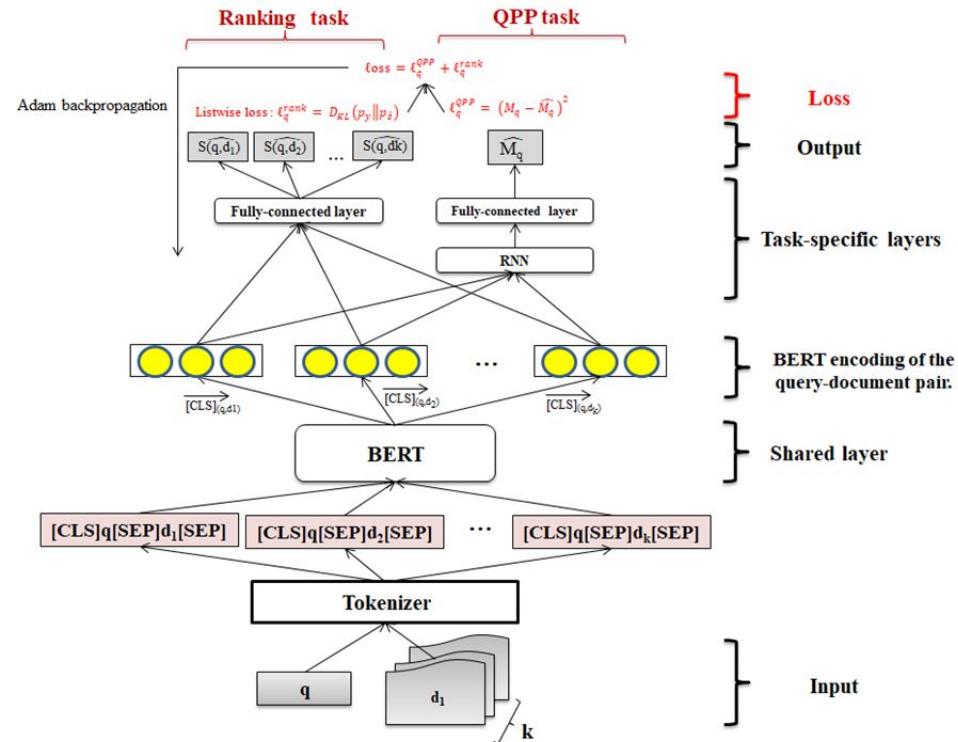
- Jointly learns to rank documents and predict the quality of the retrieved list for a given query.
- Fine-tunes a shared pre-trained BERT-based language model based on ad hoc retrieval and QPP tasks in order to capture the semantic interactions between documents and queries.



# Learning to Rank and Predict

## Multi-task Query Performance Prediction Framework (M-QPPF)

- QPP task can be viewed as a regression problem minimizing the squared error between the predicted QPP score and true performance.
- Learning the parameters of the ranking model can be accomplished using a listwise learning to rank paradigm.
- M-QPPF simultaneously optimizes two different loss functions, one loss function for the document ranking task and another for the QPP task.

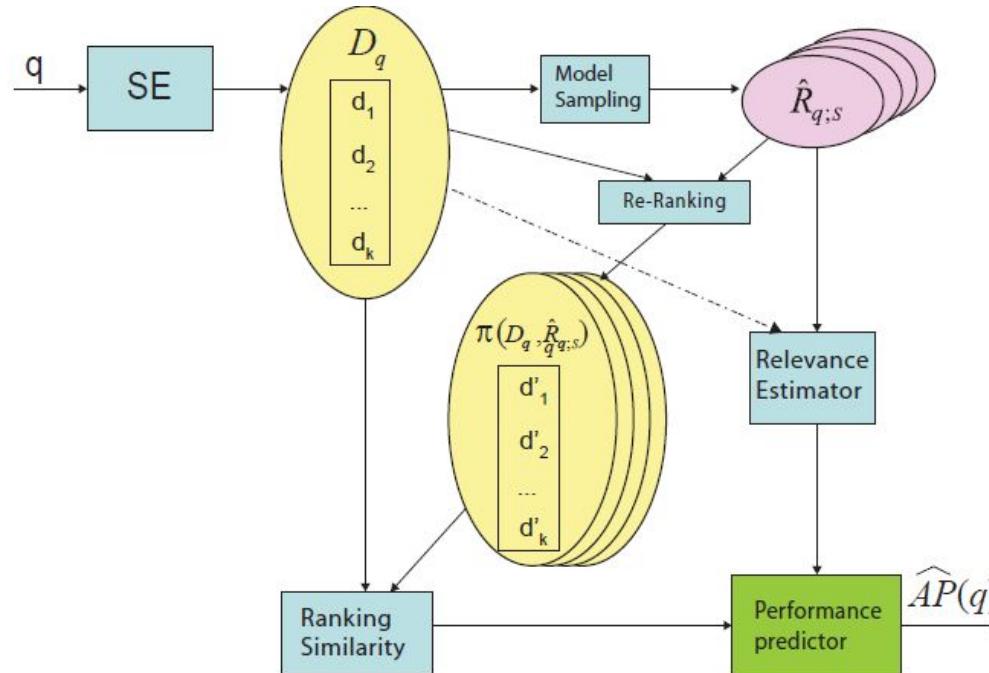


# Utility Estimation Framework

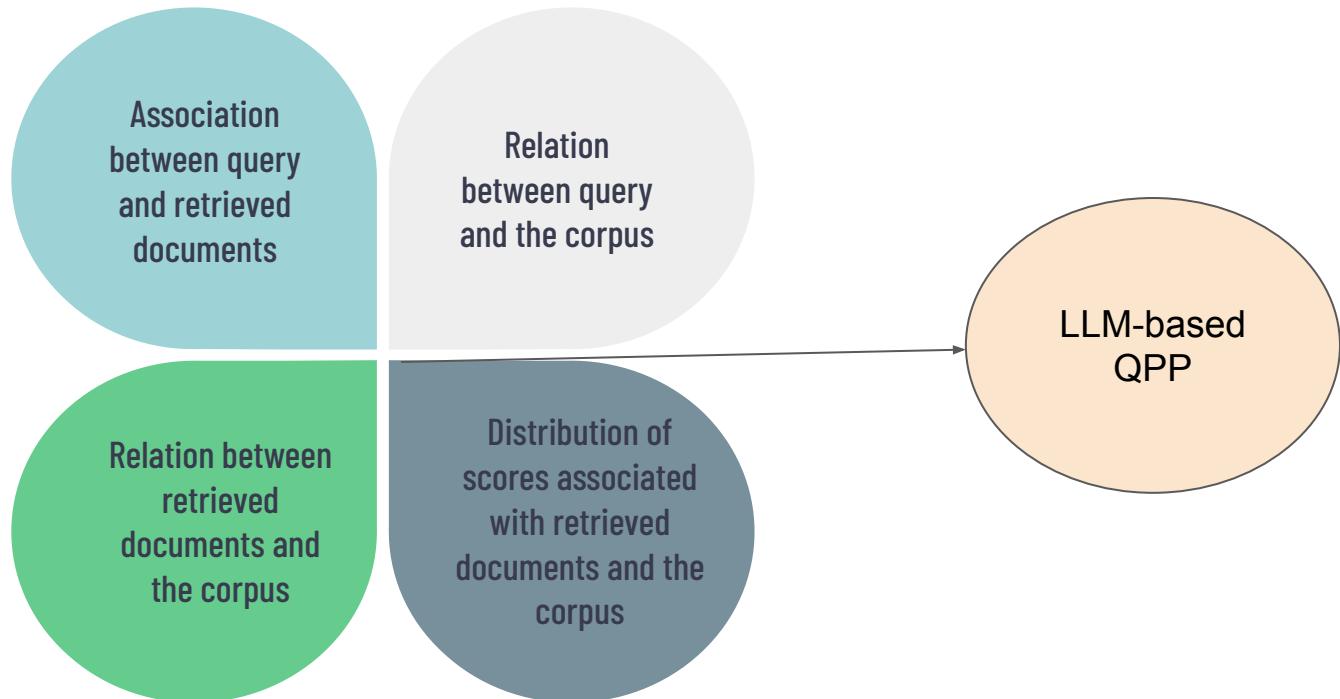
**Idea:** Integrating post-retrieval predictors based on statistical decision theory.

**Objective:** Predicting the utility a user gains from the results retrieved by a query.

**Approach:** Predicting utility as the similarity between retrieved ranked list and an ideal ranked list.

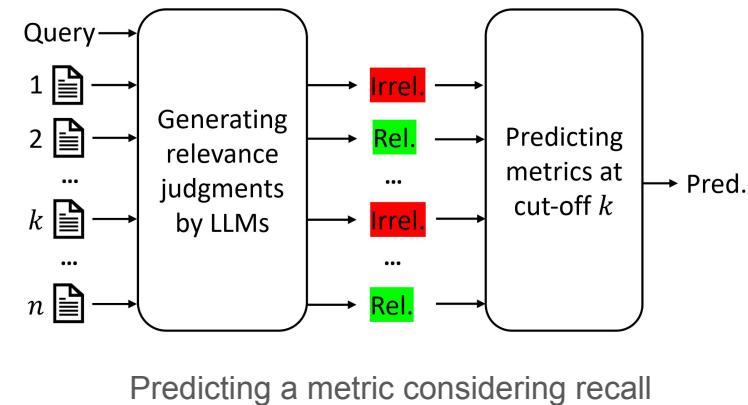
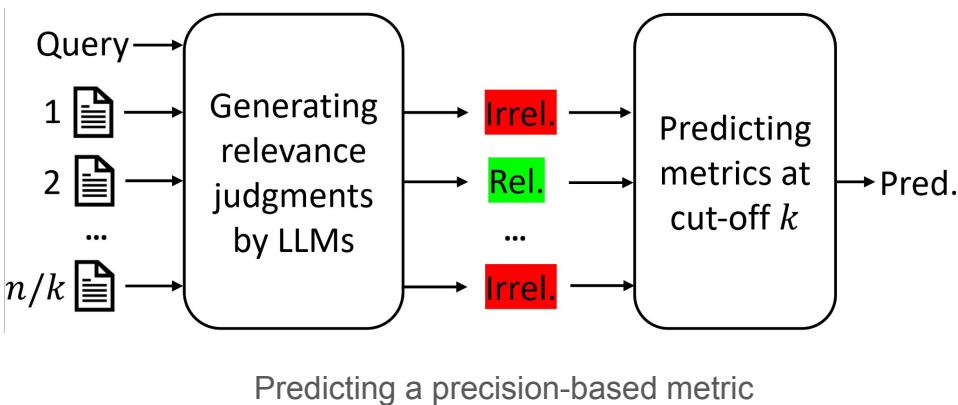


# Post retrieval QPP



# LLM-based QPP

- [1] proposes **QPP-GenRE**, which predicts IR measures using LLM-generated judgments
  - We devise an *approximation strategy* for predicting a metric considering recall
    - only judges the top  $n$  items in a ranked list, where  $n \ll \#$  documents in the corpus
    - avoids the cost of traversing the entire corpus to identify all relevant items for a query



# LLM-based QPP

- Prompting LLMs for relevance prediction yields limited and unstable performance
- [1] fine-tune LLMs for relevance prediction
  - LLMs: the Llama and Mistral families, with sizes ranging from 1B to 70B
  - Fine-tuning method: QLoRA, a parameter-efficient fine-tuning method
  - Training data: human-labeled relevance judgments of MS MARCO

**Instruction:** Please assess the relevance of the provided passage to the following question.  
Please output “Relevant” or “Irrelevant”.

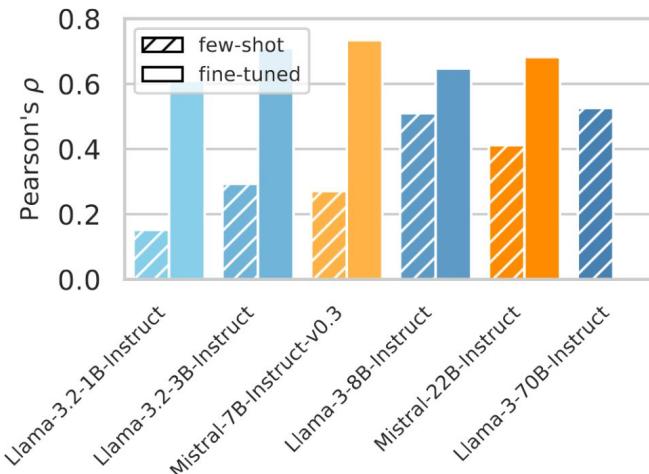
Question: {question}

Passage: {passage}

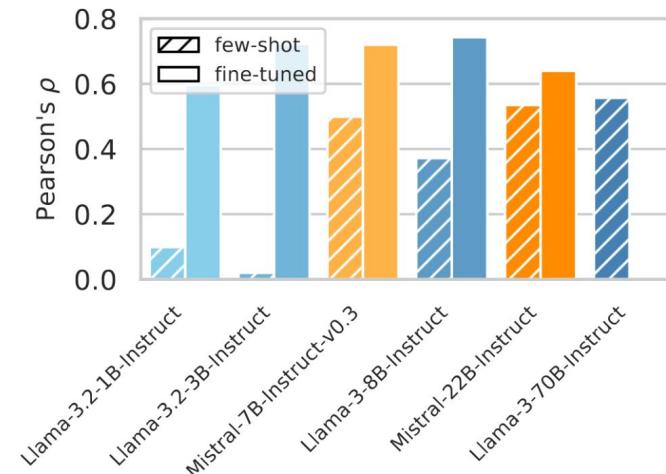
Output: Relevant/Irrelevant

# LLM-based QPP

- [1] found that
  - fine-tuning markedly improves the quality of relevance judgment generation and QPP
  - fine-tuned Llama-3.2-3B-Instruct offers the best trade-off between QPP quality and computational overhead



(a) TREC-DL 19

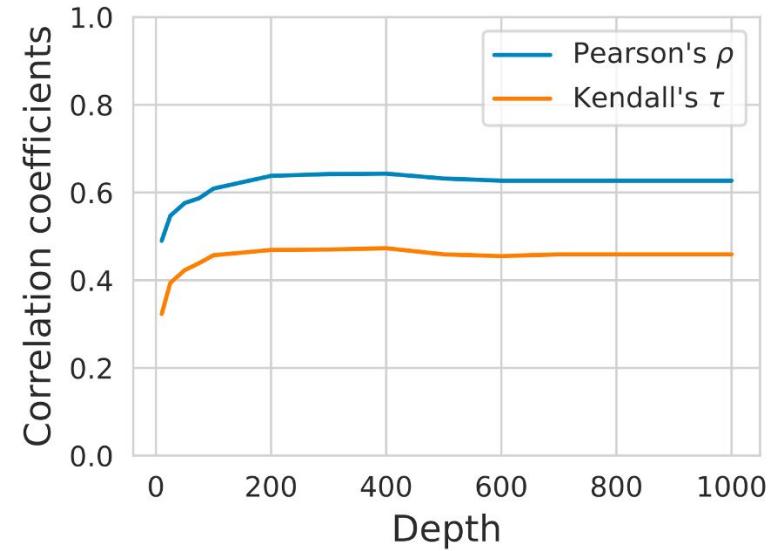


(b) TREC-DL 20

Pearson's  $\rho$  correlation coefficients between BM25' actual nDCG@10 values and those predicted by QPP-GenRE with LLMs

# LLM-based QPP

- [1] found that
  - for the proposed *approximation strategy*, judging up to 100–200 items in a ranked list suffices for predicting nDCG@10



QPP quality (predicting BM25 in terms of nDCG@10) w.r.t. judging depth

# Q & A

# QPP for various search scenarios

# QPP for various search scenarios

- QPP has been investigated in various scenarios:
  - Text search
    - Ad-hoc search
    - Conversational search
    - Open-domain question answering
  - Image search
    - Text-to-image search
    - Image-to-image search

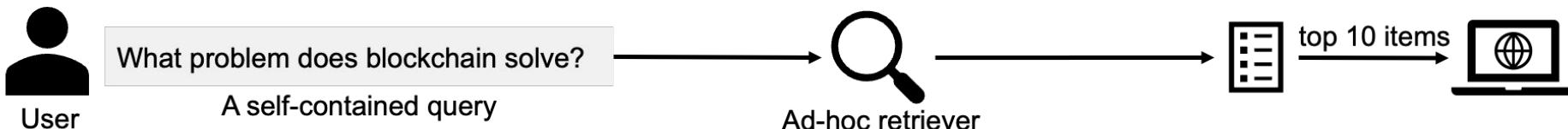
# QPP for various search scenarios

- QPP for conversational search
  - Why QPP for conversational search?
    - E.g., effective QPP could help a conversational system to decide an appropriate action to be taken at the next turn

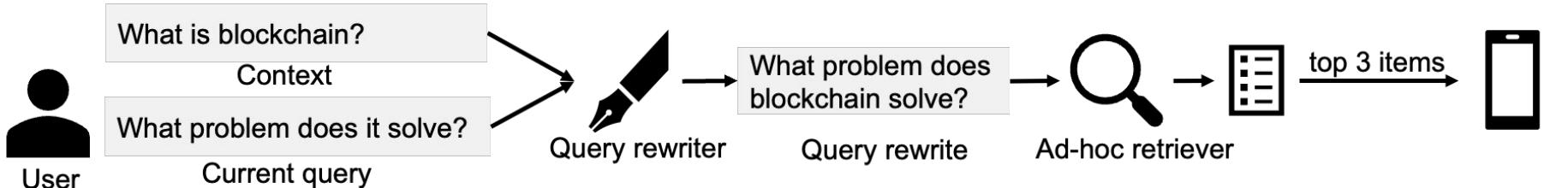
# QPP for various search scenarios

- QPP for conversational search
  - Ad-hoc search vs. conversational search
    - Self-contained vs. context-dependent queries
    - Deeper ranked list vs. only top of the ranked list

## Ad-hoc search



## Query rewriting-based retrieval

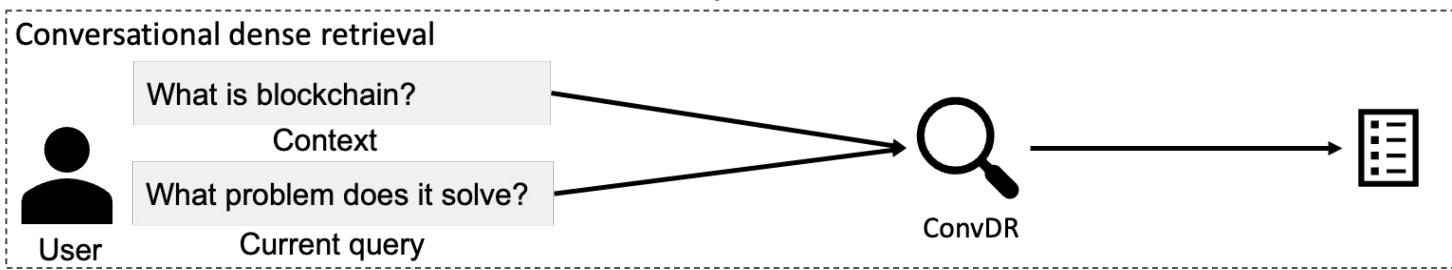


# QPP for various search scenarios

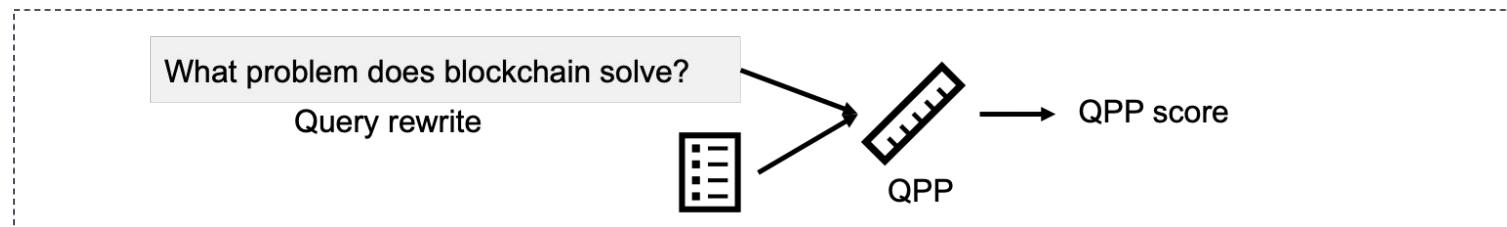
- QPP for conversational search
  - **How well QPP methods designed for ad-hoc search generalize in conversational search?**
  - [1] reproduces QPP methods in the three settings of conversational search
    - RQ1: Estimate the retrieval quality of (for top-ranked items) different query rewriting-based retrieval methods?
    - RQ2: Estimate the retrieval quality (for top-ranked items) of a conversational dense retrieval method?
    - RQ3: Estimate the retrieval quality for longer-ranked lists?

# QPP for various search scenarios

- QPP for conversational search
  - RQ2: Estimate the retrieval quality (for top-ranked items) of a conversational dense retrieval method? [1]
    - Predict the retrieval quality of ConvDR [2]



- Feed different query rewrites into QPP methods



[1] Meng et al. Query Performance Prediction: From Ad-hoc to Conversational Search. SIGIR 2023.

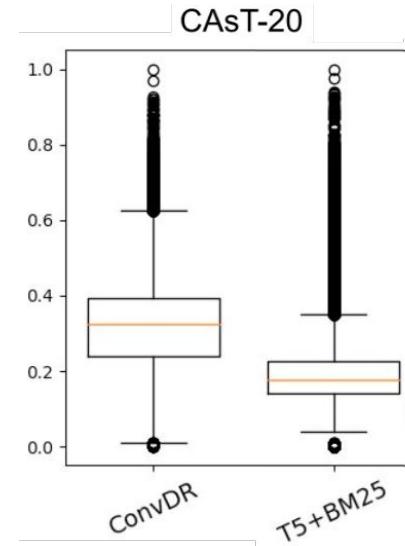
[2] Yu et al. Few-Shot Conversational Dense Retrieval. SIGIR 2021.

# QPP for various search scenarios

- QPP for conversational search
  - RQ2: Estimate the retrieval quality of (for top-ranked items) different query rewriting-based retrieval methods? [1]
    - Findings:
      - Feeding query writes works well; QPP quality tends to be better if query rewriting quality is higher
      - Supervised QPP methods achieve STOA only when having abundant training data
      - Unsupervised QPP methods are competitive in most cases, especially score-based QPP methods

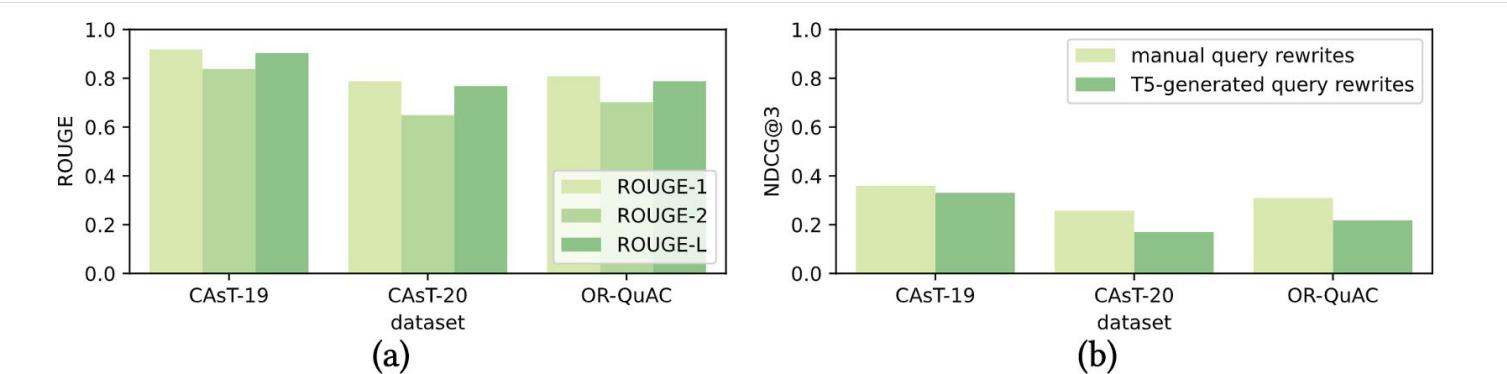
# QPP for various search scenarios

- QPP for conversational search
  - Why score-based methods exhibit a good performance? [1]
    - The ConvDR's score distribution displays a high variance
    - Score-based methods bypasses the query understanding challenge



# QPP for various search scenarios

- QPP for conversational search
  - How to improve QPP for conversational search?
    - [1] conducts an empirical analysis:
      - Lower query rewriting quality yields lower retrieval quality
      - Query rewriting quality provides evidence for QPP



**Figure 1:** The similarity between manual and T5-generated query rewrites in terms of ROUGE (a) and the retrieval quality of BM25 for manual/T5-generated query rewrites in terms of NDCG@3 (b).

# QPP for various search scenarios

- QPP for conversational search
  - How to improve QPP for conversational search?
    - [1] proposes perplexity-based QPP framework (PPL-QPP)
      - Evaluate the query rewriting quality via perplexity
      - Inject the quality into the QPP via linear interpolation
    - [1] found that
      - PPL-QPP results in higher QPP quality, especially on datasets where query rewriting is challenging

$$\text{final QPP score} = \alpha \cdot \frac{1}{\text{perplexity}} + (1 - \alpha) \cdot \text{QPP score}$$

# QPP for various search scenarios

- QPP for conversational search
  - How to improve QPP for conversational search?
    - Embeddings from conversational dense retrievers have the potential to be used for QPP
    - [1] proposes two geometric post-retrieval QPP methods
      - Fetch embeddings of query and retrieved document from conversational dense retrievers
      - Measure the proximity of the query and documents in the embedding space
      - Result in improved QPP quality

# QPP for various search scenarios

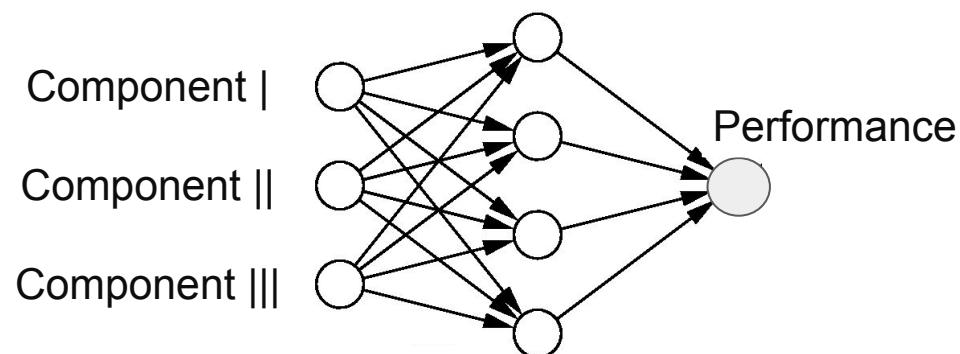
- QPP for open-domain question answering (QA)
  - Ad-hoc search vs. open-domain QA [1]
    - recall-oriented vs. precision-oriented
    - documents vs short answers
    - relevant items vs. direct answers

# QPP for various search scenarios

- QPP for open-domain question answering (QA)
  - [1] predicts the “quality” of a retrieved passage list using two parts
    - To what extent the list provide relevant items to the query
      - Post-retrieval QPP methods
    - To what extent the passages contain answers (entities)
      - The presence of named entities that may answer the question
      - Consider answer types
        - {Person, Organization, Location, Date, ...}

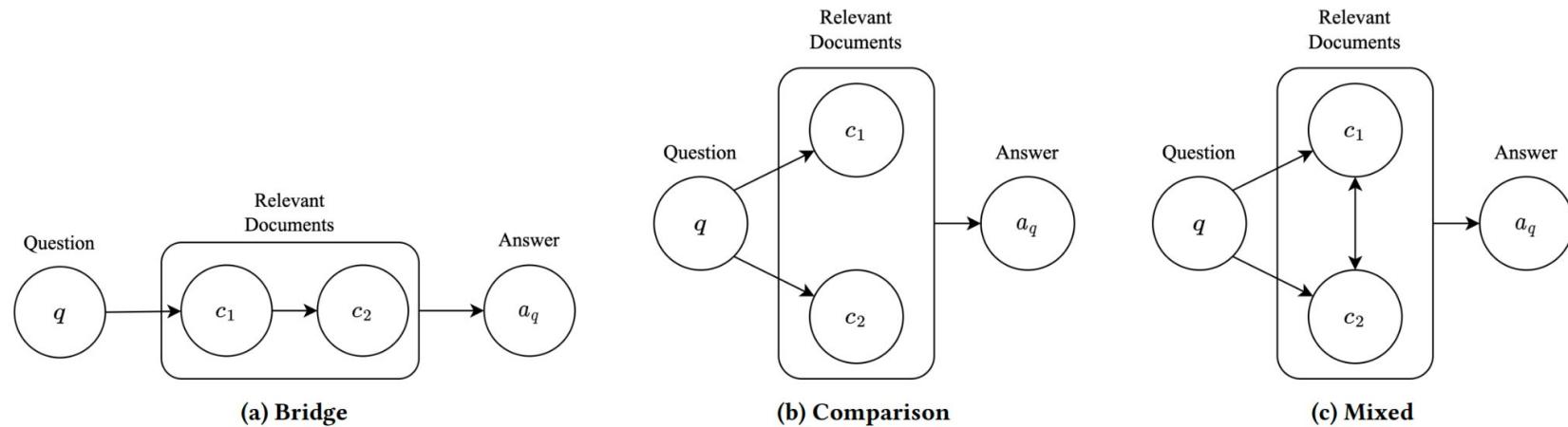
# QPP for various search scenarios

- QPP for open-domain question answering (QA)
  - [1] proposes a regression-based supervised QPP method
    - Aggregate three kinds of features:
      - Ranking scores
      - BERT(query)
      - BERT(query || answer 1), ..., BERT(query || answer k)
    - Feed them into a fully-connected network producing a single real value



# QPP for various search scenarios

- QPP for open-domain question answering (QA)
  - No research in QPP for multi-hop QA
  - [1] focuses on open-domain multi-hop QA
    - Decompose each question into a few retrieval steps
    - Estimate the difficulty of retrieving evidence under each path, using use corpus-based statistics and unsupervised QPP methods



# QPP for various search scenarios

- QPP for image search
  - QPP for text-to-image search
    - [1] proposes adapted clarity scores, which measures the difference in the distribution of the retrieved images and the whole collection
    - [1] proposes adapted coherence scores, which measures the visual similarity among the retrieved images
    - [2] reconstructs an image query based on the retrieved images, and measures the query reconstruction error

[1] Tian et al. Query difficulty prediction for web image search. In TMM 2011..

[2] Tian et al. Query difficulty estimation for image search with query reconstruction error. In TMM 2014.

# QPP for various search scenarios

- QPP for image search
  - QPP for image-to-image search
    - [1] proposes the first benchmark for query-by-example content-based image retrieval
      - Propose several pre- and post-retrieval QPP methods
      - None of the predictors achieve high performance across all data sets and retrieval methods



# Q & A

# Applications of QPP

# Applications of QPP

- QPP has been applied to various downstream scenarios:
  - Query-oriented
    - Query variant selection
    - Clarifying question selection
    - Selective query expansion
  - Ranker-oriented
    - Ranking quality improvement
    - IR system configuration selection
    - Ranker selection
    - Fusion-based retrieval
    - Candidate generation
  - Others
    - Action prediction
    - Conversation contextualization
    - Query routing
    - Query-specific pool depth prediction

# Applications of QPP

- Query variant selection
  - It is impossible to find the most effective query variant by running all of variants, especially in systematic reviews
  - [1,2,3] use QPP methods to select the best-performing query variant or sort query variants, given the same information need and ranker
    - QPP methods predict the difficulty of query variations given the same topic worse than predicting topic difficulty

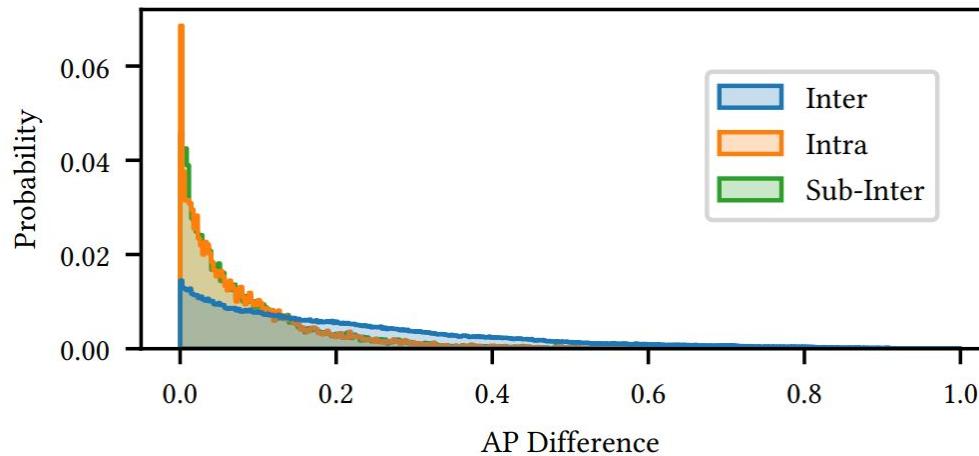
[1] Thomas et al. Tasks, Queries, and Rankers in Pre-Retrieval Performance Prediction. ADCS 2017.

[2] Scells et al. Query Variation Performance Prediction for Systematic Reviews. SIGIR 2018.

[3] Di Nunzio et al. Study of a Gain Based Approach for Query Aspects in Recall Oriented Tasks. Applied Sciences 2021.

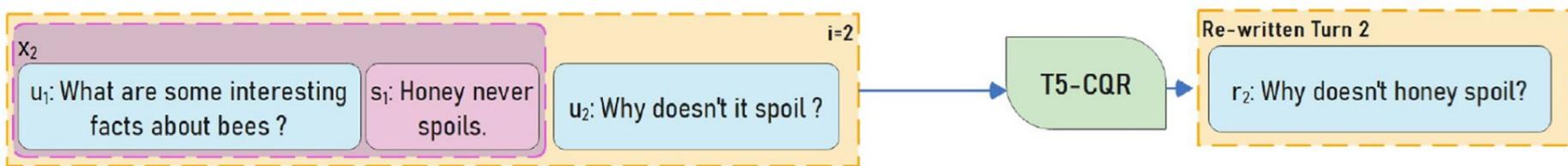
# Applications of QPP

- Query variant selection
  - [1] reveals the reason:
    - Actual effectiveness differences among query variants are smaller than those among topics.



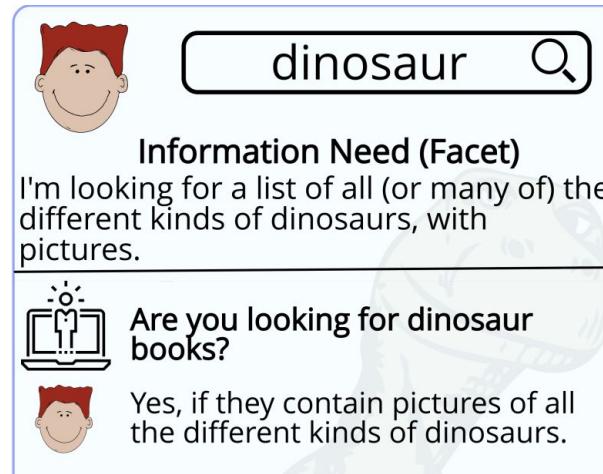
# Applications of QPP

- Query variant selection
  - In conversational search, query rewrites can be generated using various sources
  - [1] uses QPP select the better query rewrite from different ones
    - Compare the QPP scores for different query rewrites; the one with higher score is used for ranking
    - Significantly improve ranking performance compared to scenarios without selection



# Applications of QPP

- Clarifying question selection
  - In conversational search, selecting a clarifying question that helps to clarify users' initial query from a large question bank is challenging [1,2]



[1] Aliannejadi et al. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. SIGIR 2019.

[2] Hashemi et al. Guided Transformer: Leveraging Multiple External Sources for Representation Learning in Conversational Search. SIGIR 2020.

# Applications of QPP

- Clarifying question selection
  - [1,2] directly use a score-based QPP method to predict the ranking quality for each candidate clarifying question and select the one with the maximum predicted ranking quality
    - Result in higher ranking quality compared to not asking questions
    - Result in comparable ranking quality compared to learning-to-rank methods

Method	Qulac-T Dataset				
	MRR	P@1	nDCG@1	nDCG@5	nDCG@20
OriginalQuery	0.2715	0.1842	0.1381	0.1451	0.1470
$\sigma$ -QPP	0.3570	0.2548	0.1960	0.1938	0.1812
LambdaMART	0.3558	0.2537	0.1945	0.1940	0.1796

- [1] also regards a QPP value as a feature and feed it into a neural-based clarifying question selection method

[1] Aliannejadi et al. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. SIGIR 2019.

[2] Hashemi et al. Guided Transformer: Leveraging Multiple External Sources for Representation Learning in Conversational Search. SIGIR 2020.

# Applications of QPP

- Selective query expansion (selective relevance feedback)
  - Query expansion improves average ranking quality but degrade ranking quality for certain queries [1,2]
  - [1] sets a threshold for the clarity score for an initial ranking result
    - it can well identify bad-to-expand queries
  - [2] follows [1] but use qppBERT-PL scores

[1] Cronen-Townsend et al. A Language Modeling Framework for Selective Query Expansion. Technical Report 2004.

[2] Datta et al. A Deep Learning Approach for Selective Relevance Feedback. ECIR 2024.

# Applications of QPP

- Selective query expansion (conversational search)
  - Some queries in conversational search contain omissions, coreferences, or ambiguities
  - [1] uses score-based QPP method to determine whether the current query should be expanded with keywords from the previous turns
    - Regard the maximum BM25 ranking score as the QPP score
    - Set a threshold for the QPP score
    - Generally more effective than always doing query expansion

---

QPP	R@1000	MAP	NDCG@3
✓	0.730	0.211	0.259
	0.728	0.207	0.264

# Applications of QPP

- QPP has been applied to various downstream scenarios:
  - Query-oriented
    - Query variant selection
    - Clarifying question selection
    - Selective query expansion
  - Ranker-oriented
    - Ranking quality improvement
    - IR system configuration selection
    - Ranker selection
    - Fusion-based retrieval
    - Candidate generation
  - Others
    - Action prediction
    - Conversation contextualization
    - Query routing
    - Query-specific pool depth prediction

# Applications of QPP

- Ranking quality improvement [1]
  - Use QPP scores as features for learning-to-rank models
  - QPP features show promise in increasing effectiveness

Source	Class	Feature	Description	Total
query	QPP	AvICTF [4]	Pre-retrieval performance predictor	1
query	QPP	AvIDF [4]	Pre-retrieval performance predictor	1
query	QPP	AvPMI [4]	Pre-retrieval performance predictor	1
query	QPP	EnIDF [4]	Pre-retrieval performance predictor	1
query	QPP	Gamma1 [4]	Pre-retrieval performance predictor	1
query	QPP	Gamma2 [4]	Pre-retrieval performance predictor	1
query	QPP	TermCount	Number of unique terms	1
query	QPP	TokenCount	Number of tokens	1
query	QPP	N-GramScore	Likelihood of ngram query in anchor or title fields	8
query	QCI	AcronymSenses	Number of acronym senses	1
documents	QCI	WPDisambSenses [19]	Number of disambiguation senses per document	18
documents	QCI	WPDisambCount	Number of disambiguation pages retrieved	6
query	QCI	EntityCount	Number of named entities in the query	4
query	QLM	NGramScore	Likelihood of ngram query in query log	3
clicks	QLM	ClickCount	Number of clicks	3
clicks	QLM	ClickEntropy [5]	Click entropy at the URL level	1
clicks	QLM	HostEntropy [22]	Click entropy at the host level	1
clicks	QLM	ResultCount	Number of displayed results in a session	3
clicks	QLM	SessionDuration	Session duration in seconds	3
documents	QTC	DocEntityCount	Number of retrieved entities (products, persons, organisation, locations)	18
documents	QTC	DocEntityEntropy	Entity entropy of centroid document	18
documents	QTC	DocEntityPairwiseCosine	Entity distance over pairs of top documents	54
documents	QTC	WPCategoryCount	Number of retrieved categories	6
documents	QTC	WPCategoryEntropy	Category entropy of centroid document	6
documents	QTC	WPPairwiseCosine	Categorical distance over pairs of top documents	18
<b>TOTAL</b>				178

Class	Feature	NDCC@20
Baseline (no query features)		0.2832
QPP		
Rank 1	Gamma1	0.3033 <sup>▲</sup>
Rank 2	TermCount	0.30217 <sup>△</sup>
Rank 3	1-GramScore on anchor field	0.29857
Rank 4	4-GramScore on anchor field	0.29727
Mean	-	0.2955 (2/16)

# Applications of QPP

- IR system configuration selection [1,2]
  - IR systems' performance impacted by numerous parameters, leading to a huge number of possible combinations of parameter values
  - Individual queries need different treatments.

**Table 1: Description of the system parameters that we use to build our dataset**

Parameter	Description & values <sup>2</sup>
Retrieval model	21 different retrieval models: DirichletLM, JsKLS, BB2, PL2, DFRee, DFIO, XSqrAM, DLH13, HiemstraLM, InL2, DLH, DPH, IFB2, TFIDF, InB2, InexpB2 , DFRBM25 , BM25, LGD, LemurTFIDF, InexpC2.
Expansion model	7 query expansion models: nil, Rocchio, KL, Bo1, Bo2, KLCorrect, Information, KLComplete.
Expansion documents	Number of documents used for query expansion: 2, 5, 10, 20, 50, 100.
Expansion terms	Number of expansion terms: 2, 5, 10, 15, 20.
Expansion min-docs	Minimal number of documents an expansion term should appear in: 2, 5, 10, 20, 50.

[1] Deveaud et al. Learning to Adaptively Rank Document Retrieval System Configurations. TOIS 2018

[2] Deveaud et al. Learning to Rank System Configurations. CIKM 2016.

# Applications of QPP

- IR system configuration selection
  - [1,2] formulate it as a learning-to-rank problem
    - Regard possible system configurations as candidates and use learning to rank to select an appropriate configuration for a given query
    - Consider QPP scores as query statistical features
    - Show that query statistical features produces variable effects

Group	Variants	Features
QUERYSTATS	3 Pre-retrieval features with mean and standard deviation variants of IDF 40 Letor features with mean and standard deviation variants (0 stands for Title, 1 for Body and 2 for both) 3 Query difficulty predictors	IDF [38, 40], and CLARITY [24].  SFM(DL,0/1/2), SFM(TF,0/1/2), SFM(IDF,0/1/2), SFM(SUM_TF,0/1/2), SFM(MEAN_TF,0/1/2), SFM(TF_IDF,0/1/2), SFM(BM25,0/1/2), SFM(LMIR_DIR,0/1/2), SFM(LMIR_JM. $\lambda$ -C-0.4,0/1/2), Pagerank_prior, Pagerank_rank  WIG [82], QF [82], and NQC [68].
QUERYLING	12 WordNet features with mean and standard deviation variants 18 Linguistic query features No variant	SYNONYMS, HYPOONYMS, MERONYMS, HOLONYMS, HYPERNYMS, and SISTER- TERMS [57].  NBWORDS, INTERR, NP, ACRO, NUM, PREP, CC, PP, VBCONJ, UNKNOWN, AVGSIZE, AVGMORPH, %CONSTR, AVGSYNSETS, SYNTDEPTHAVG, SYNTDEPTHMAX, SYNTDISTANCEAVG, and SYNTDISTANCEMAX [58].
RETRMODEL	1 feature representing retrieval model	Retrieval model such as HiemstraLM, BM25, and so on (see Table 1)
EXPANSION	4 features for query expansion	Expansion model, number of expansion documents, number of expansion terms, and minimum number of documents (see Table 1).

[1] Deveaud et al. Learning to Adaptively Rank Document Retrieval System Configurations. TOIS 2018

[2] Deveaud et al. Learning to Rank System Configurations. CIKM 2016.

# Applications of QPP

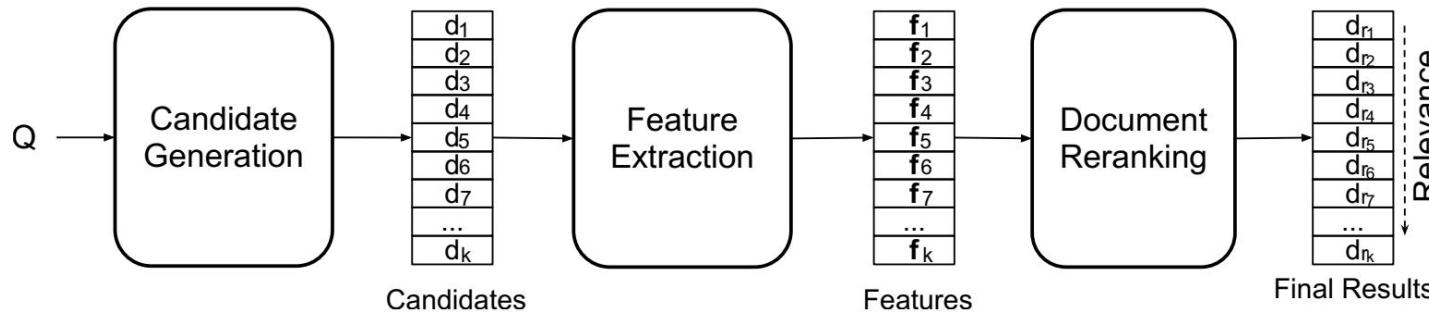
- Ranker selection
  - Select the appropriate ranker for a new test corpus from a ranker pool
  - [1] utilizes a bunch of QPP methods to rank the performance of dense retrievers for a new test corpus
    - Score-based QPP methods perform poorly because retrieval scores are not normalized across dense retrievers
    - Reference list-based QPP method perform better

# Applications of QPP

- Fusion-based retrieval
  - Given multiple retrieved lists, they should have weights that reflect their retrieval quality with respect to the query
  - [1] uses score-based QPP methods to predict list weights
    - Retrieval results using QPP weights are worse than a naive baseline (use a ranker's actual performance on the training set as the list weight)
    - QPP are designed for estimating for which queries a ranker would perform better, not for comparing rankers' performance for a query

# Applications of QPP

- Candidate generation
  - Candidate generation (first-stage retrieval) is a time-consuming part in multi-stage ranking systems [1]
  - Increase efficiency without significantly reducing overall effectiveness
    - For a easy query, return less documents
    - For a hard query, return more documents



# Applications of QPP

- Candidate generation
  - [1] combines 7 pre-retrieval QPP methods to determine the parameters of the candidate generation algorithm on a per-query basis
    - For a query, compare the estimated effectiveness with a threshold to make a decision
    - QPP can keep effectiveness while improving efficiency in a conservative manner

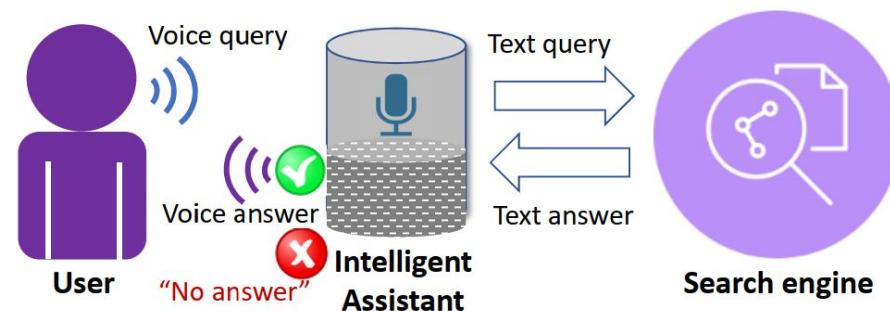
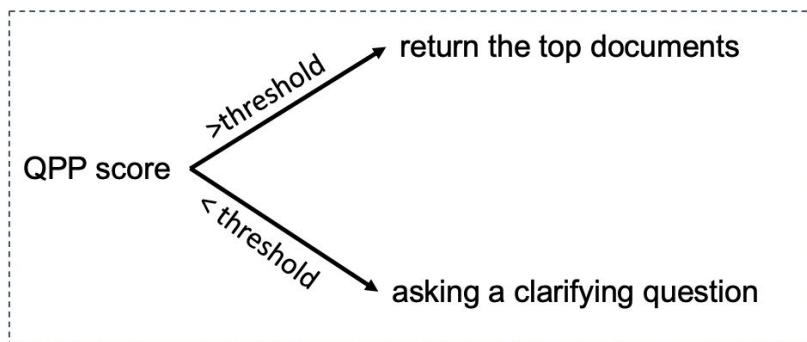
$$\text{SELECT}_{QPP}(\text{PREDICT}(q)) = \begin{cases} \{20, 2\} & \text{if } \widehat{E}(q) > \epsilon \\ \{1000, 1\} & \text{otherwise} \end{cases}$$

# Applications of QPP

- QPP has been applied to various downstream scenarios:
  - Query-oriented
    - Query variant selection
    - Clarifying question selection
    - Selective query expansion
  - Ranker-oriented
    - Ranking quality improvement
    - IR system configuration selection
    - Ranker selection
    - Fusion-based retrieval
    - Candidate generation
  - Others
    - Action prediction
    - Conversation contextualization
    - Query routing
    - Query-specific pool depth prediction

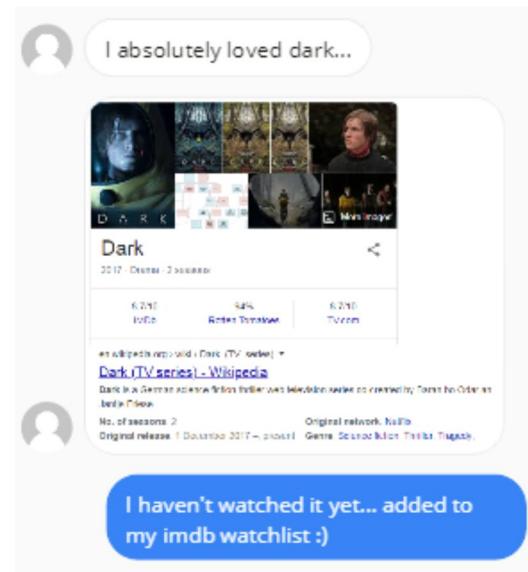
# Applications of QPP

- Action prediction in conversational search
  - When not to give answers to users?
  - [1] use score-based QPP values to predict the difficulty of a user query and use a threshold for decision
    - performance is comparable to fine-tuned BERT
  - [2] use a set of QPP features to train a classifier
    - QPP features make a difference



# Applications of QPP

- Conversation contextualization
  - Retrieve background information for the content in a conversation that is potentially difficult to comprehend [1]



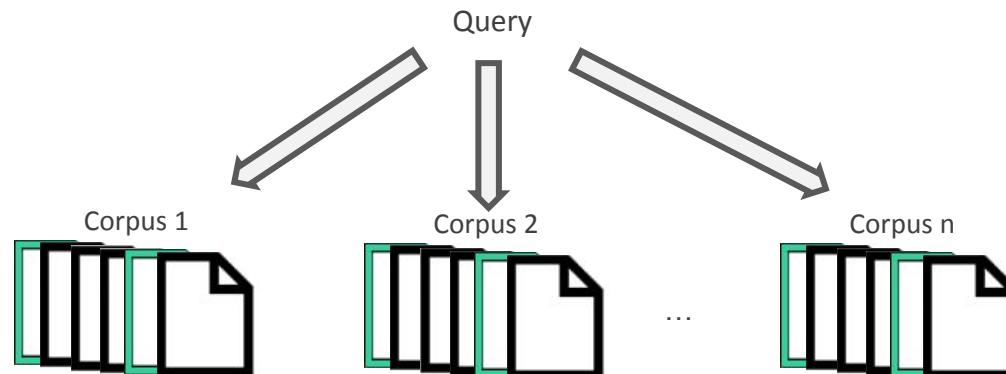
# Applications of QPP

- Conversation contextualization
  - [1] regards a text segment in a conversation as a query and use QPP methods to predict the ranking quality
    - Assume that the higher the predicted quality, the greater need for contextualization
    - QPP methods can effectively identify the text segment that needs contextualization, leading to the better performance of retrieving information relevant to the given conversation

	Term selection	$\phi$ (specificity)	$k$	BLEU	Jaccard
Baseline	Term-level	Avg idf	4	0.1459	0.0585
Ours	Window-based	Avg idf	5	<b>0.1623</b>	<b>0.0716</b>
	Window-based	NQC	4	0.1113	0.0482

# Applications of QPP

- Query routing [1]
  - In the context of multiple and distributed document repositories, route a query to the repository that can best answer the query, potentially improving ranking efficiency and effectiveness



# Applications of QPP

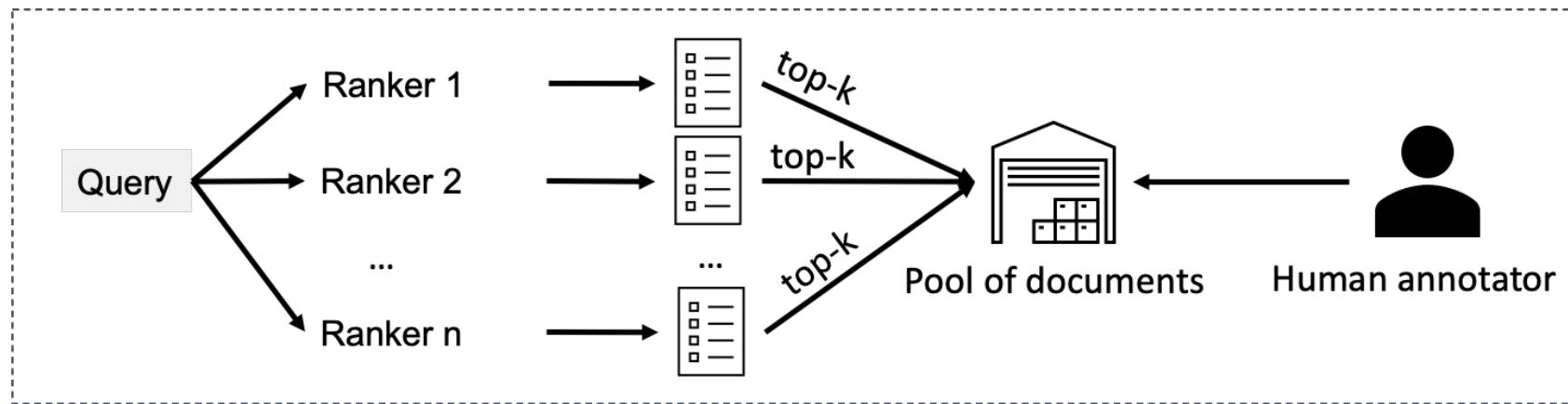
- Query routing
  - [1] builds a SVM classifier using QPP scores as features for query routing; experiments with 5 repositories show:
    - The classifier accurately routes queries to the correct repository
    - Retrieval on the repository chosen by the classifier results in higher retrieval quality than retrieval on all repositories

TABLE 12. Impact on retrieval performance when using SVM classification for query routing.

Query source	Mean average precision when query is routed to	
	Integrated repository	Domain-specific repository by SVM predictor
CACM	0.1593	0.1812
CISI	0.1019	0.1266
CRAN	0.0077	0.0071
TIME	0.6177	0.6325
TREC9	0.2755	0.2783

# Applications of QPP

- Query-specific pool depth prediction [1]
  - The common ground for relevance judgments is to use a constant depth across all queries
  - Constant depth wastes annotation budget on queries needing fewer judgments



# Applications of QPP

- Query-specific pool depth prediction
  - [1] proposes to use QPP as a variable pool depth predictor
    - Two methods based on QPP scores:
      - Inverse linear dependence
      - Linear dependence
    - Experiments:
      - Reflect the relative performance of rankers with a smaller annotation effort
      - There is no clear winner between these two methods

# Q & A

# Conclusions and future directions

# Conclusions and future directions

- Conclusion
  - What is QPP
  - QPP methods: from foundational to cutting-edge
    - Pre-retrieval
    - Post-retrieval
  - QPP for various search scenarios
    - QPP for text-based search
      - QPP for conversational search
      - QPP for open-domain QA
    - QPP for image-based search
  - QPP's applications
    - Query-oriented
    - Ranker-oriented
    - Other

# Conclusions and future directions

- More focus is needed on the following directions:
  - Predicting the performance of
    - LLM-based retrievers/re-rankers
    - generative AI systems
  - Leveraging the capabilities of LLMs to enhance QPP quality
  - Applying QPP to benefit various downstream tasks.
  - Exploring QPP in the context of multi-modal content
  - Exploring multilingual QPP

# Call for paper!

- **The QPP++ 2025 workshop (full day): Query Performance Prediction and its Applications in the Era of Large Language Models**
  - Co-located with the 47th European Conference on Information Retrieval (ECIR 2025), 6th–10th April 2025, Lucca, Italy



- Paper types:
  - original papers (4–10 pages)
  - accepted papers (1-page abstract)
- Important dates:
  - Submission DDL: 24th Feb, 2025
  - Workshop: 10th April, 2025



Website

# Thank you

# Discussions