Query Performance Prediction: From Fundamentals to Advanced Techniques

Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi & Ebrahim Bagheri

Tutorial at The 46th European Conference on Information Retrieval (ECIR ’24)
March, 2024

UNIVERSITY OF WATERLOO
UNIVERSITY OF AMSTERDAM

ECIR 2024 GLASGOW

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*.

Introduction

What is a difficult query?

- Poorly-performed
  - E.g., performance < specific threshold τ

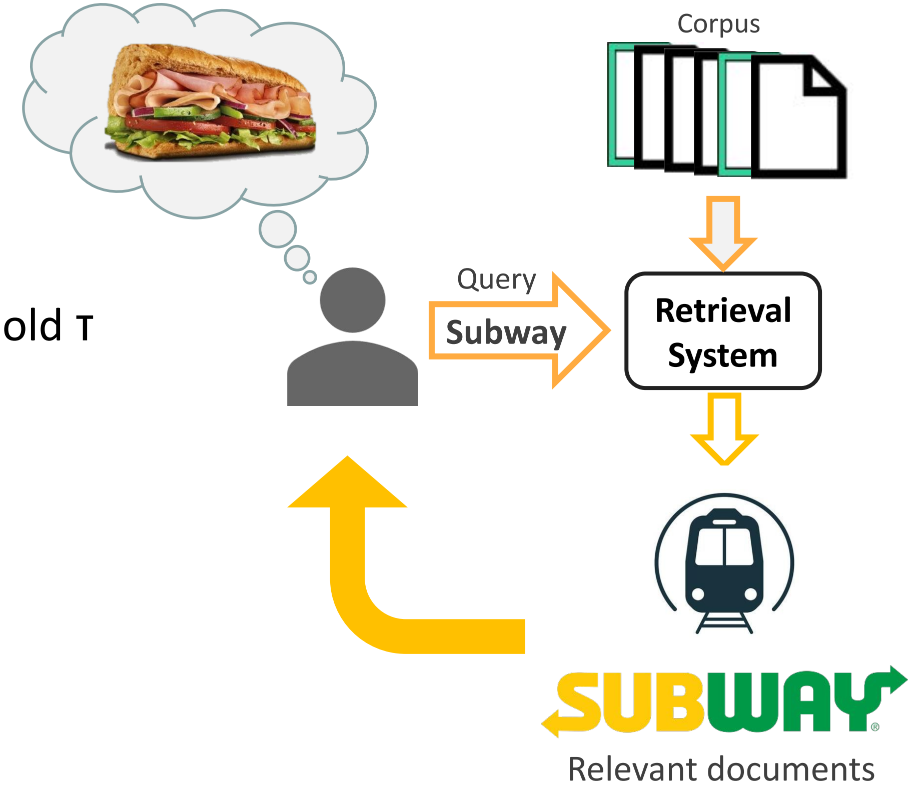
Introduction

What is a difficult query?

- ➢ Poorly-performed
  - ○ E.g., performance < specific threshold τ

Why is a query “difficult”?

1. Query term ambiguity



Introduction

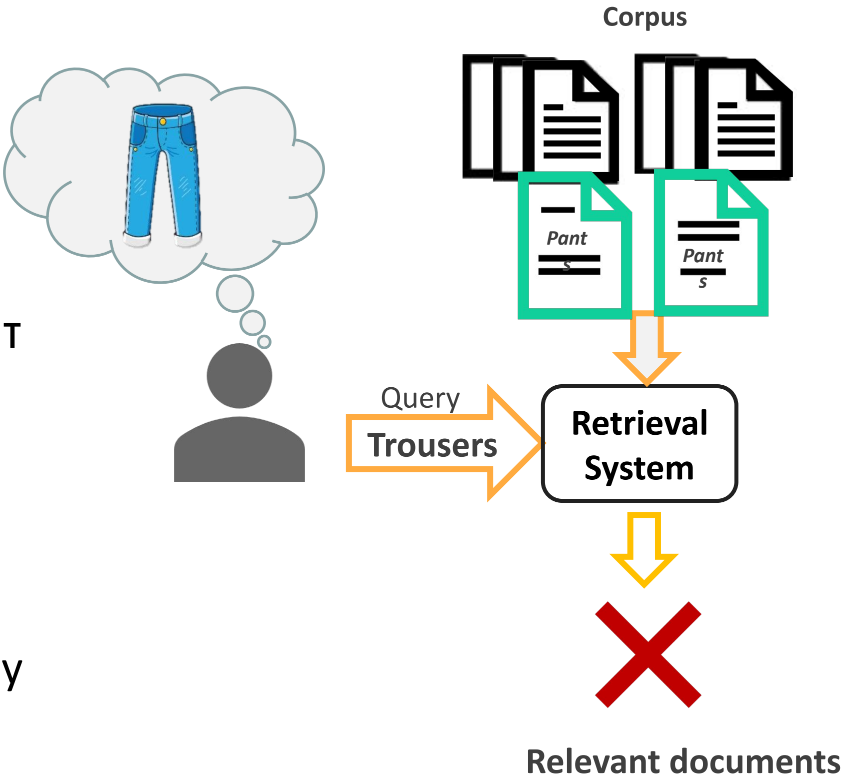
What is a difficult query?

- ➢ Poorly-performed
  - ○ E.g., performance < specific threshold τ

Why is a query “difficult”?

1. Query term ambiguity

2. Query and document language inconsistency



Corpus

Pant s

Pant s

Query

Trousers

Retrieval System

Relevant documents

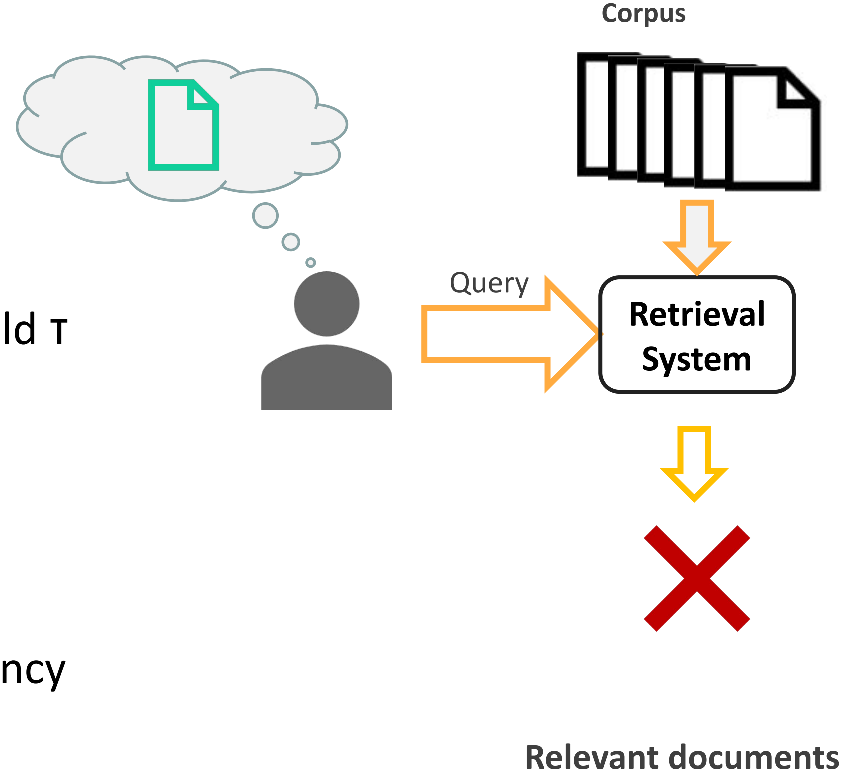
Introduction

What is a difficult query?

- ➢ Poorly-performed
  - ○ E.g., performance < specific threshold τ

Why is a query “difficult”?

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



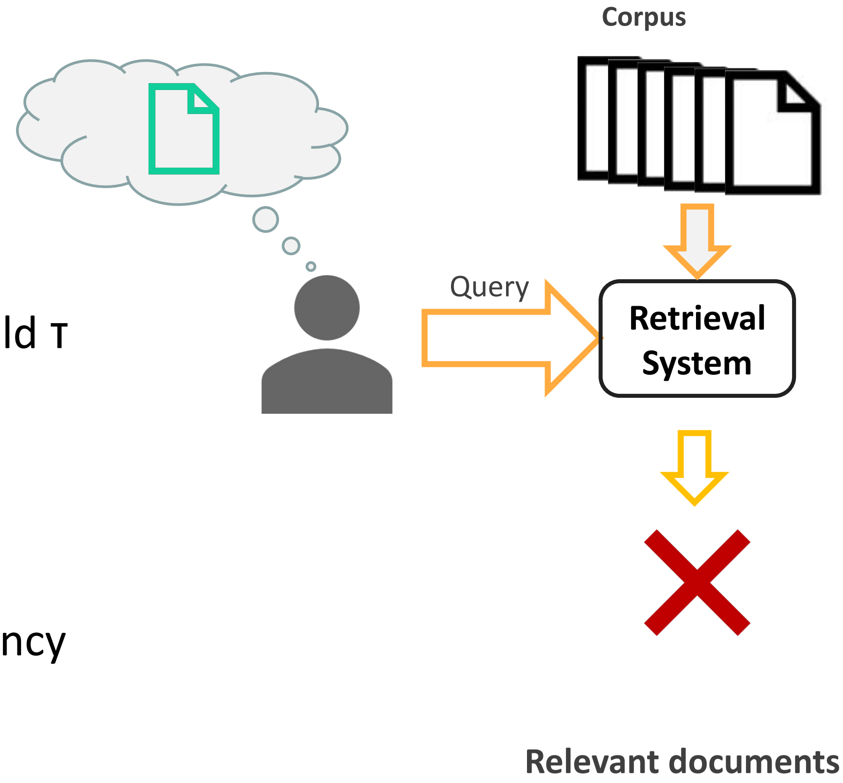
Introduction

What is a difficult query?

- ➢ Poorly-performed
  - ○ E.g., performance < specific threshold τ

Why is a query “difficult”?

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



Introduction

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 μ has to estimate the Average Precision of q , AP (q):

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

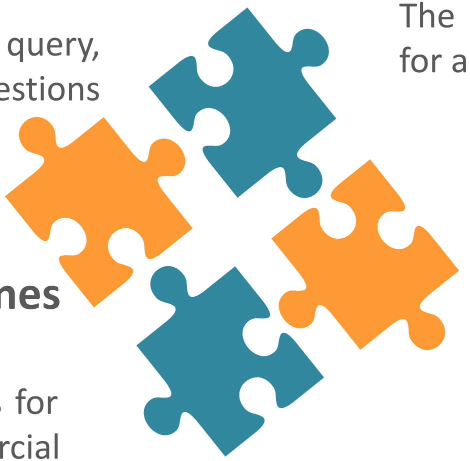
Primary Applications

Feedback to users

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

Feedback to system Administrator

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



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.

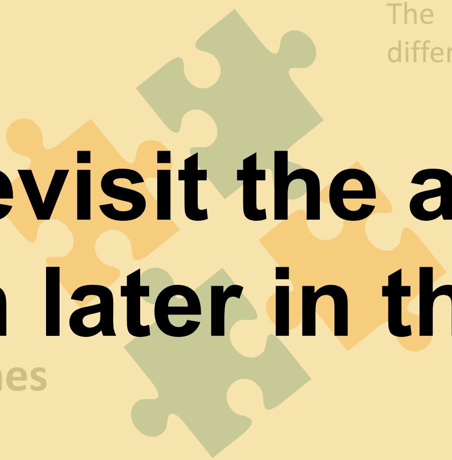
Information Retrieval administrator

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

Carmel et al. “Estimating the query difficulty for information retrieval”

Primary Applications

We will revisit the applications in depth later in this tutorial



Feedback to users

The user can rephrase the query

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

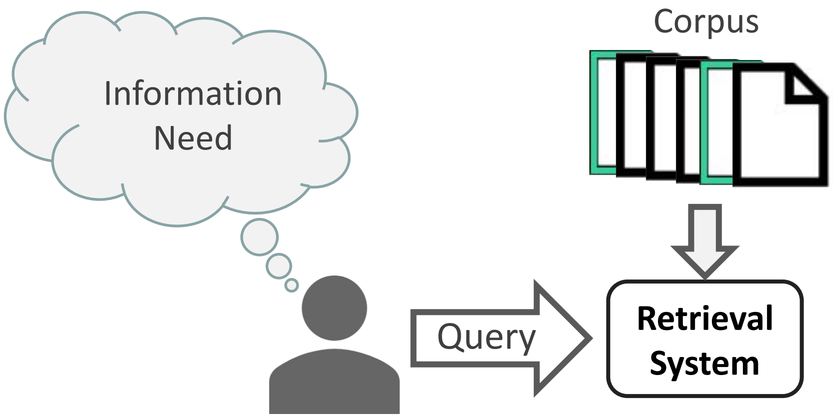
The search engine can use different strategies for a different query.

Information Retrieval administrator

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

Categories

Pre-retrieval

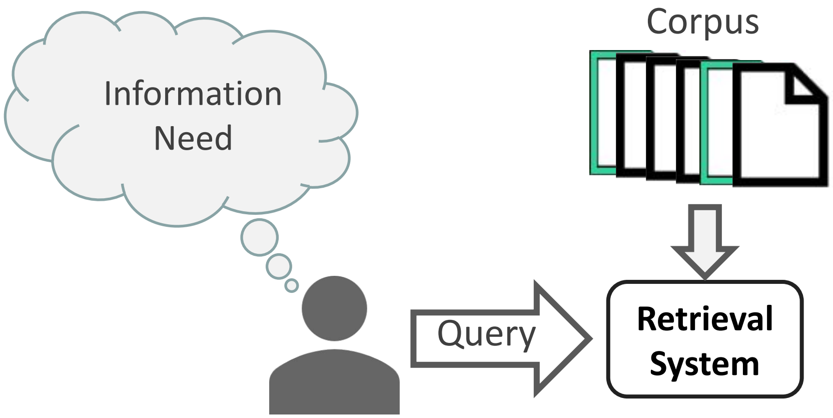


No access to retrieved Items

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

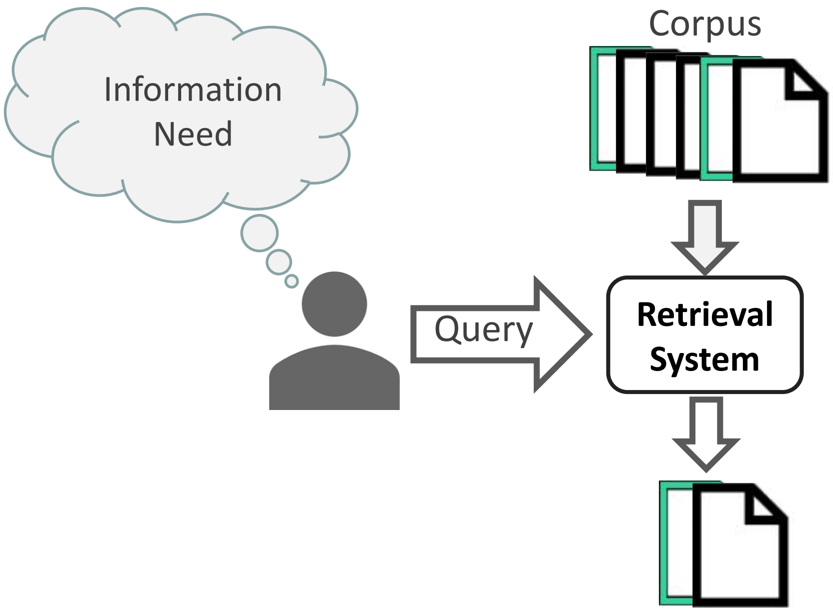
Categories

Pre-retrieval VS Post-retrieval



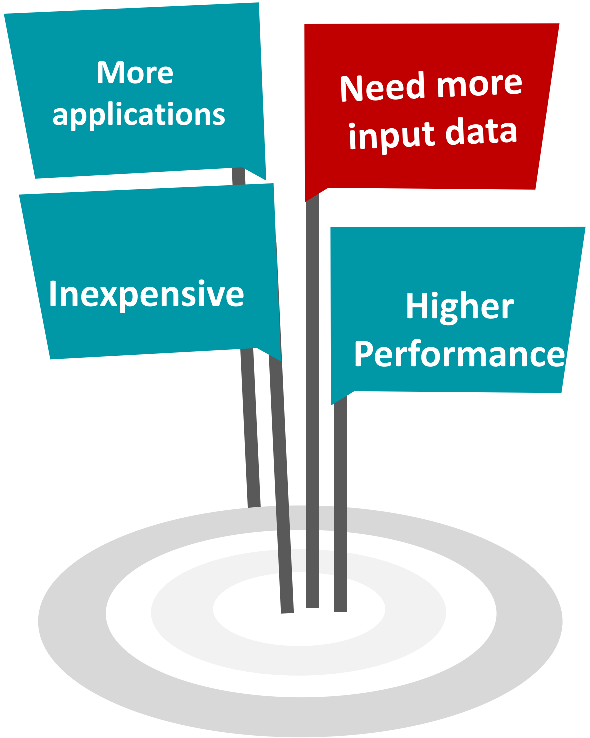
No access to retrieved Items

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



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

Categories



Post-retrieval QPP

Pre-retrieval QPP

QPP Evaluation

QPP evaluation

Given:

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

Predictor μ 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?

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

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

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 ϱ
  - Rank-based: Kendall's τ, Spearman's ϱ

[1] Ellen et al. Information Technology: The Sixth Text REtrieval Conference (TREC-6).

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)

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

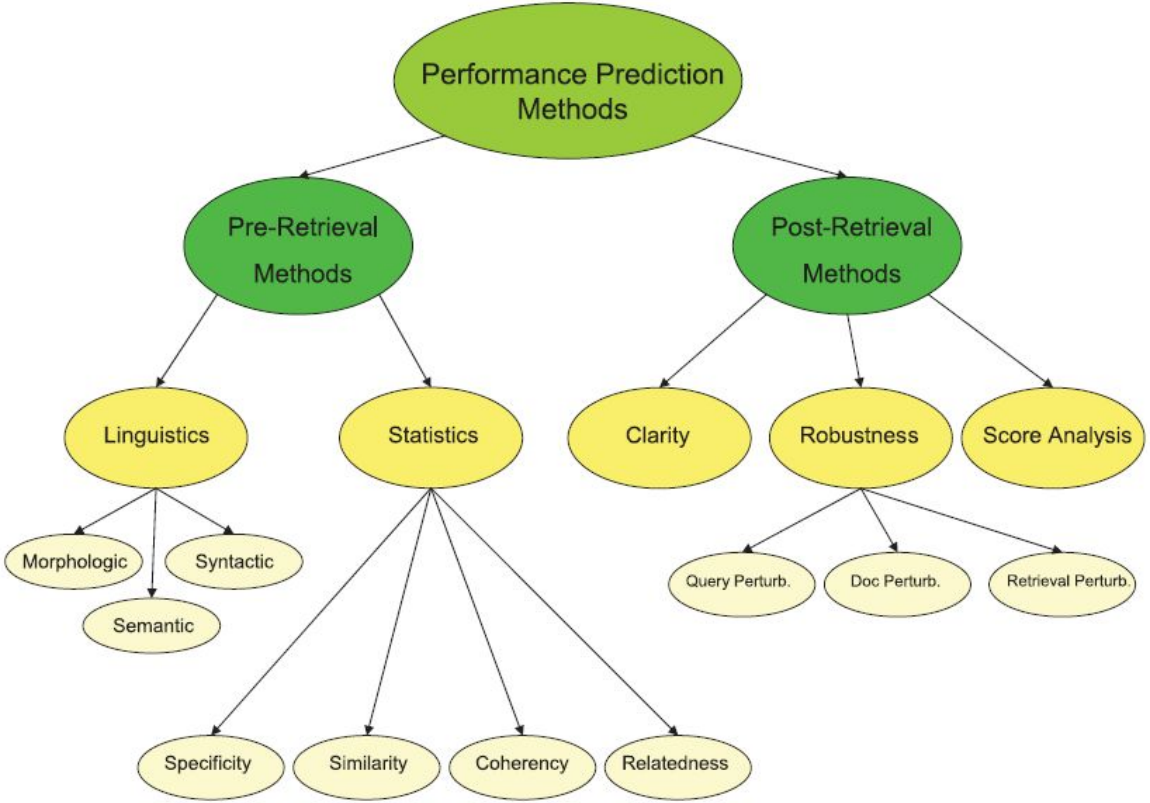
  - scaled Mean Absolute Rank Error (sMARE)

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

[1] Faggioli et al. sMARE: a new paradigm to evaluate and understand query performance prediction methods. Information Retrieval Journal.

QPP Categorization

Categories



Carmel et al. “Estimating the query difficulty for information retrieval”

Pre-retrieval QPP

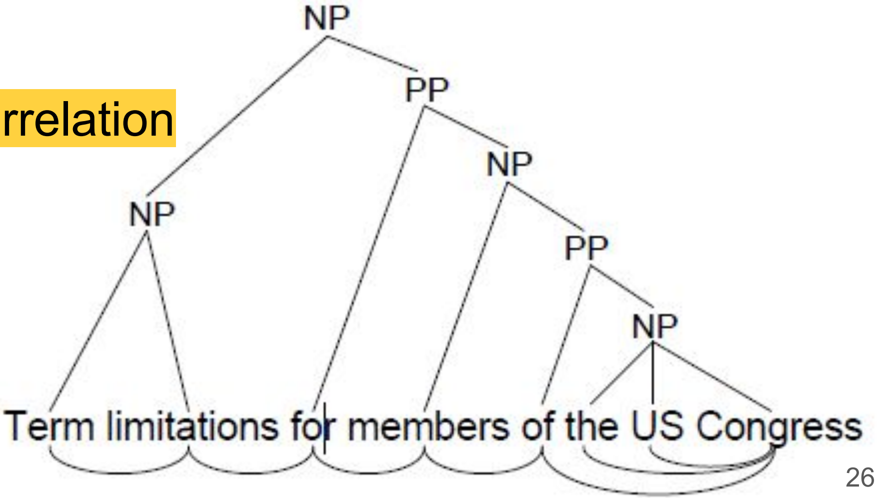
Linguistic approaches

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.

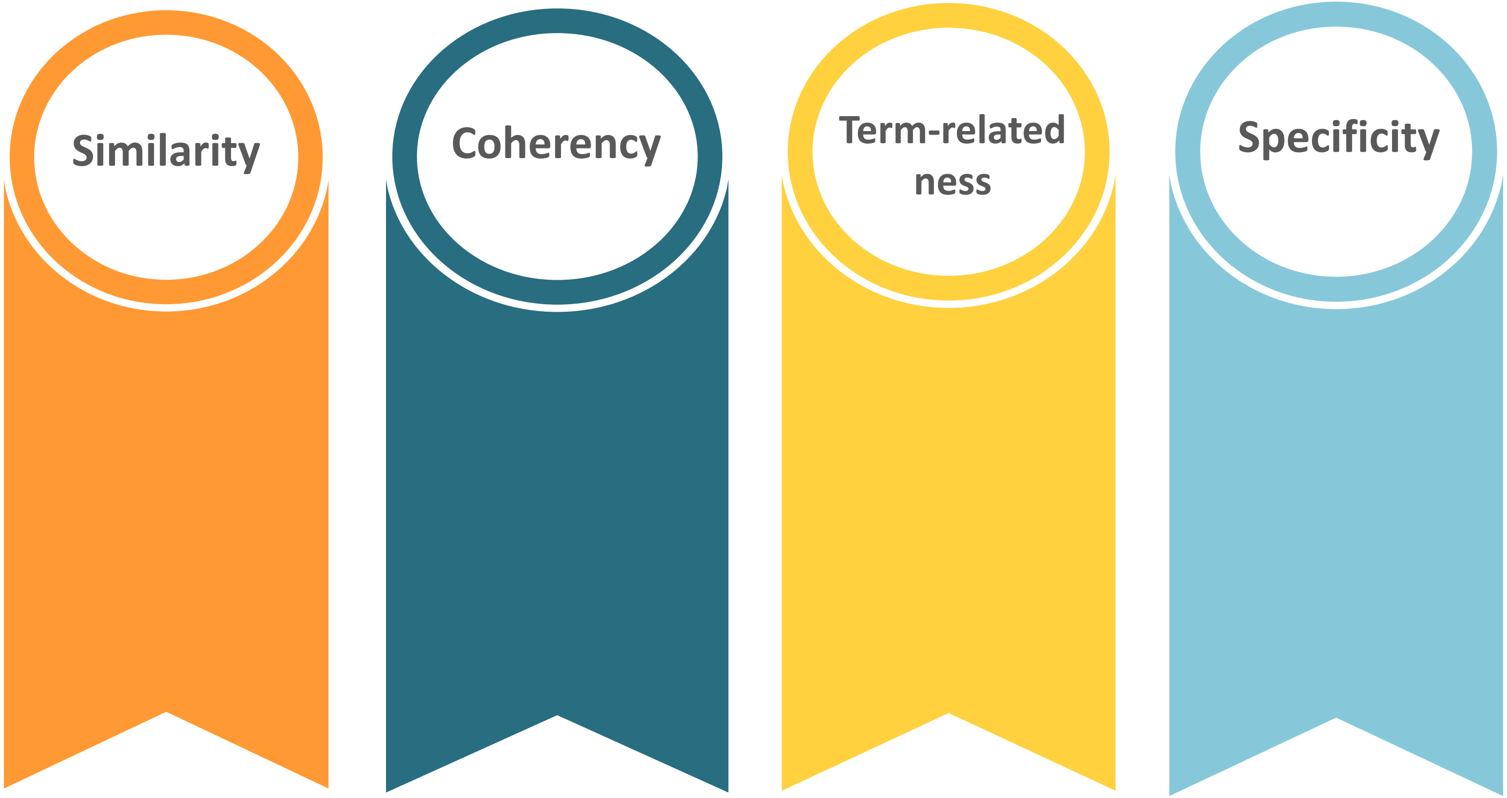
Most linguistic features showed weak or no correlation with system performance.



Dreilinger et al. "Experiences with selecting search engines using meta-search"
Mothe et al. "Linguistic features to predict query difficulty."

Statistical approaches

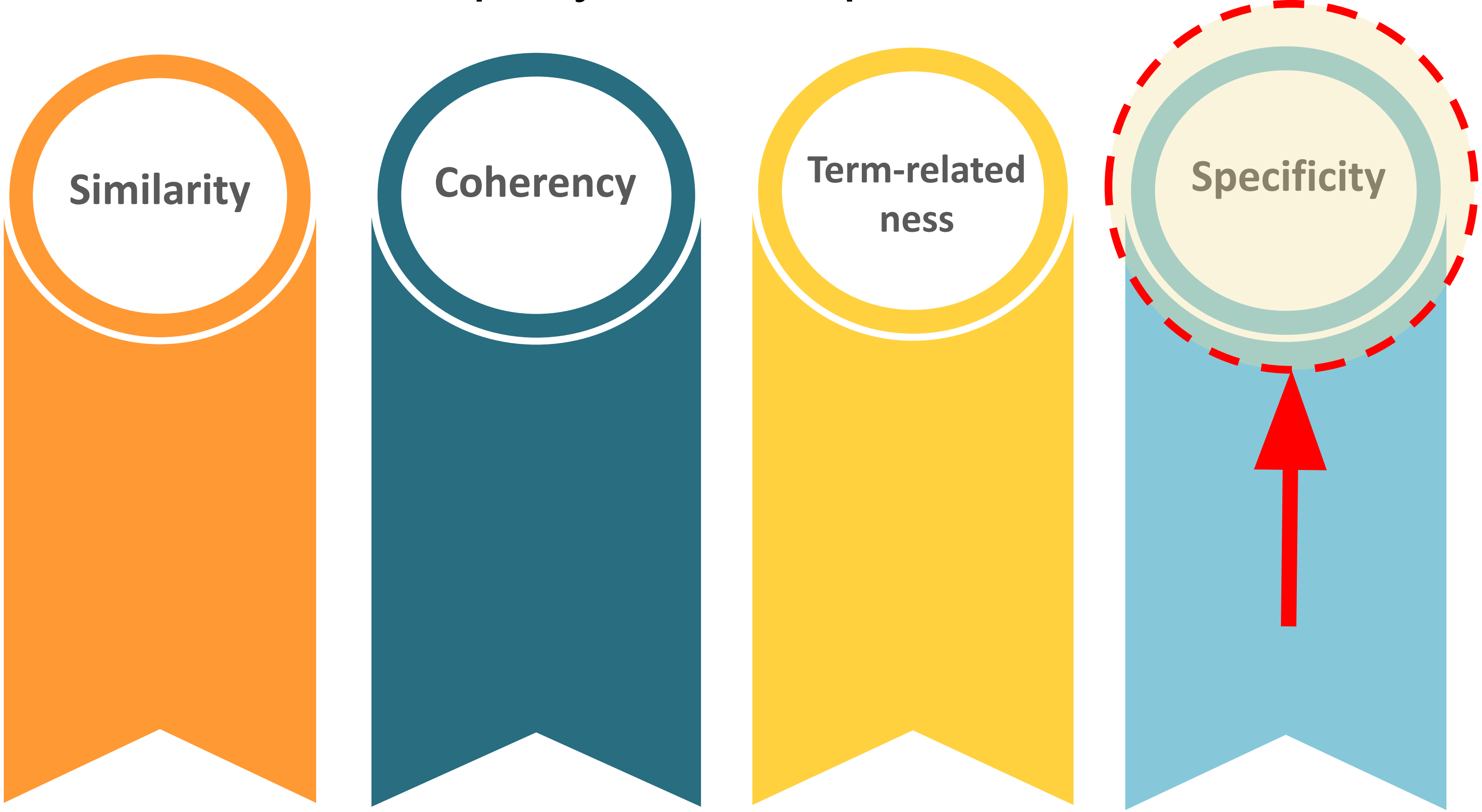
Intuition: Distribution of query term frequencies within the collection



- Similarity
- Coherency
- Term-relatedness
- Specificity

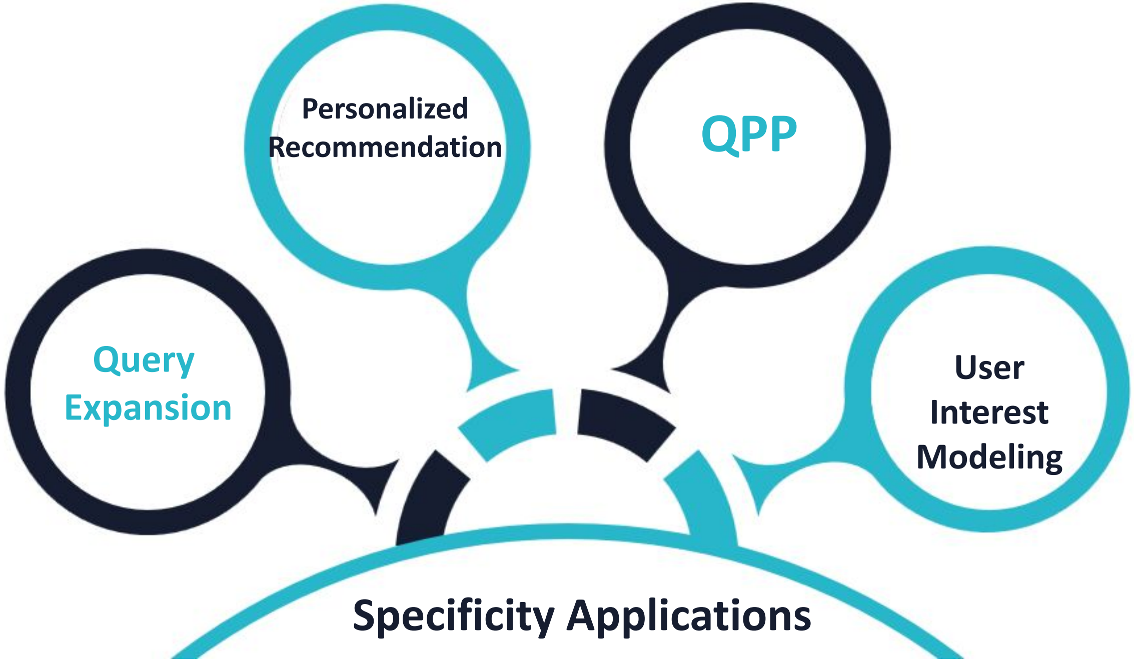
Statistical approaches

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.

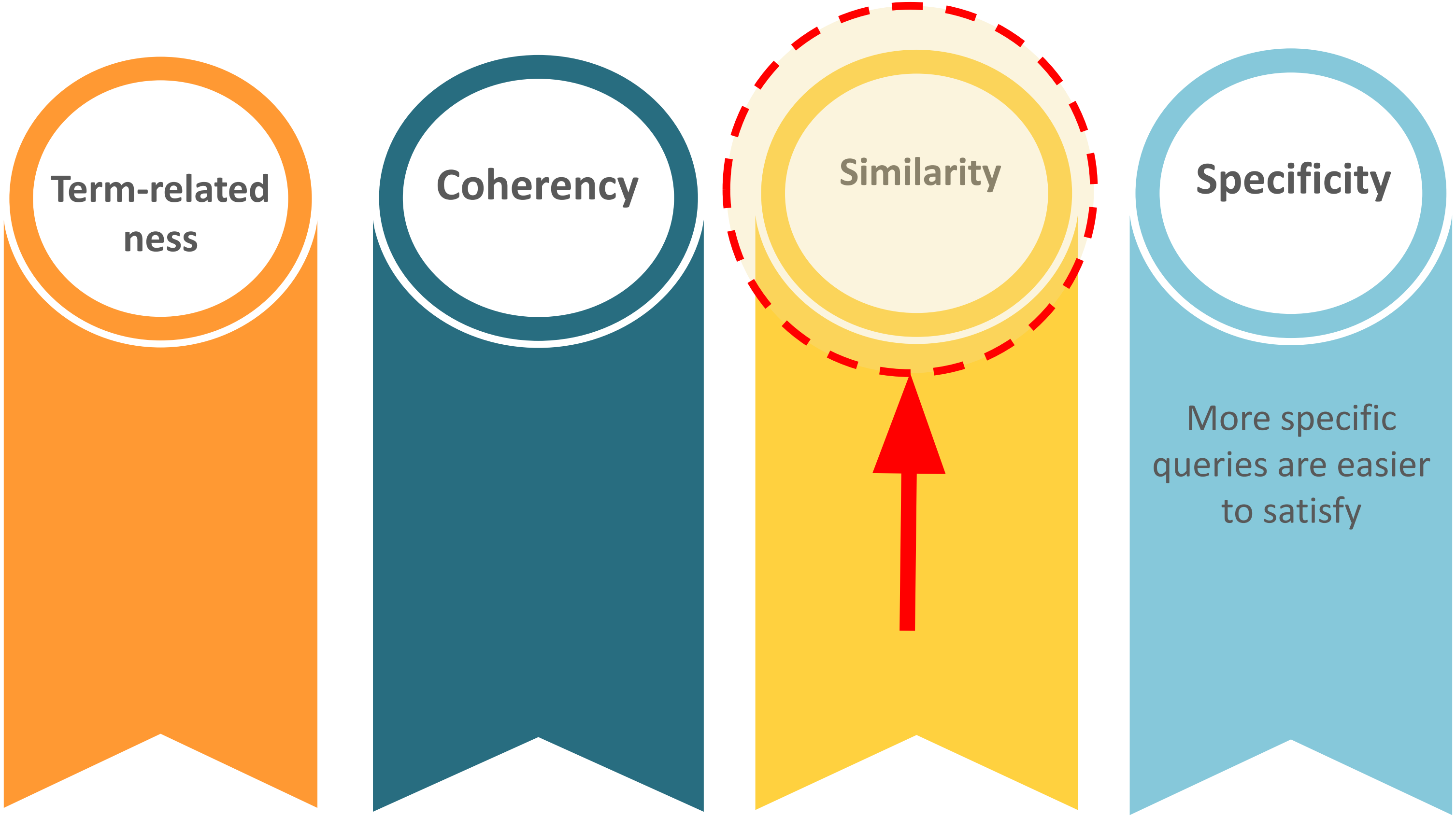
Specificity-based QPP -QS

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



Term-related ness

Coherency

Similarity

Specificity

More specific queries are easier to satisfy

Similarity-based QPP

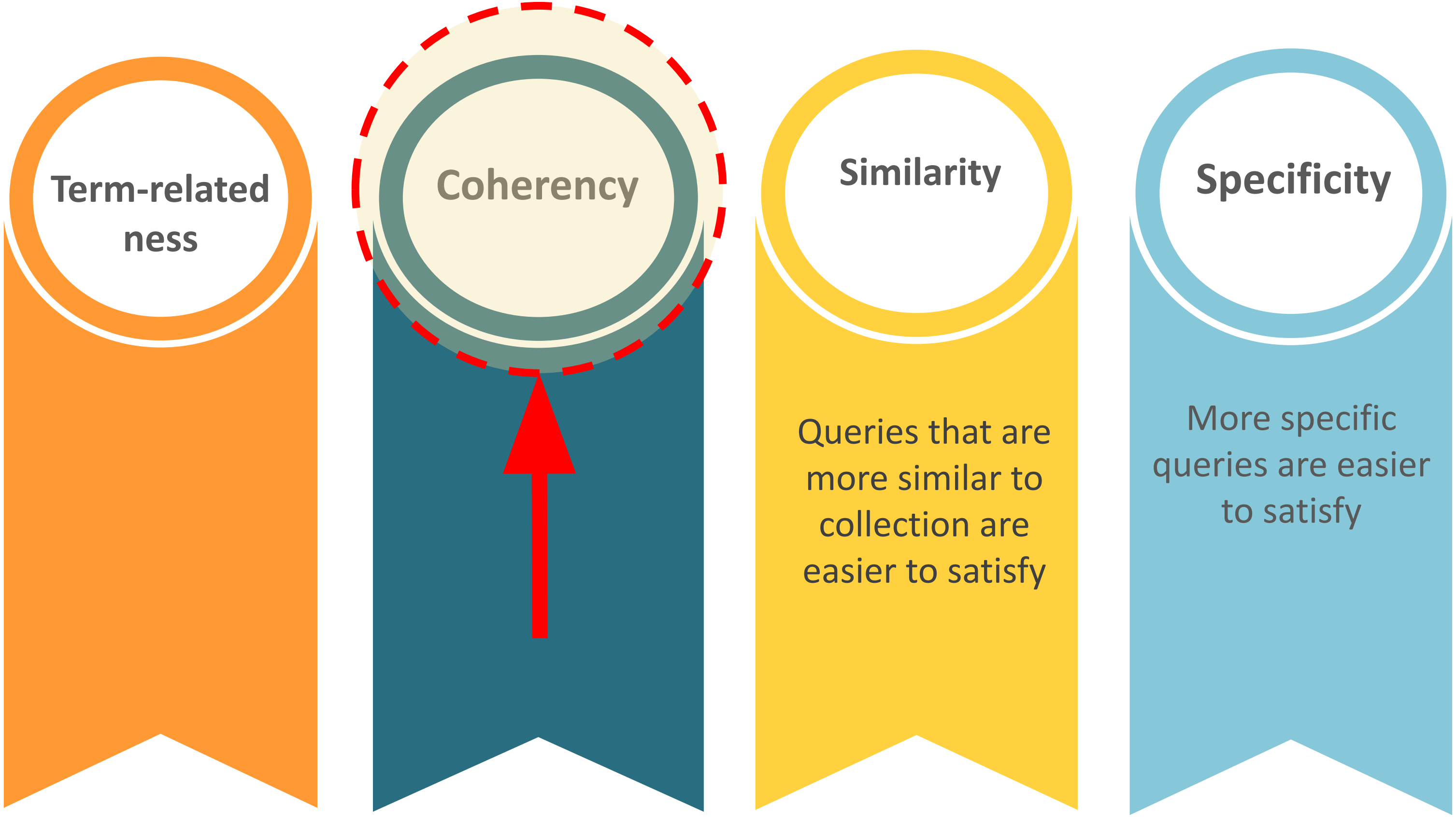
Idea: Similarity of the query and collection.

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



Queries that are more similar to collection are easier to satisfy

More specific queries are easier to satisfy

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

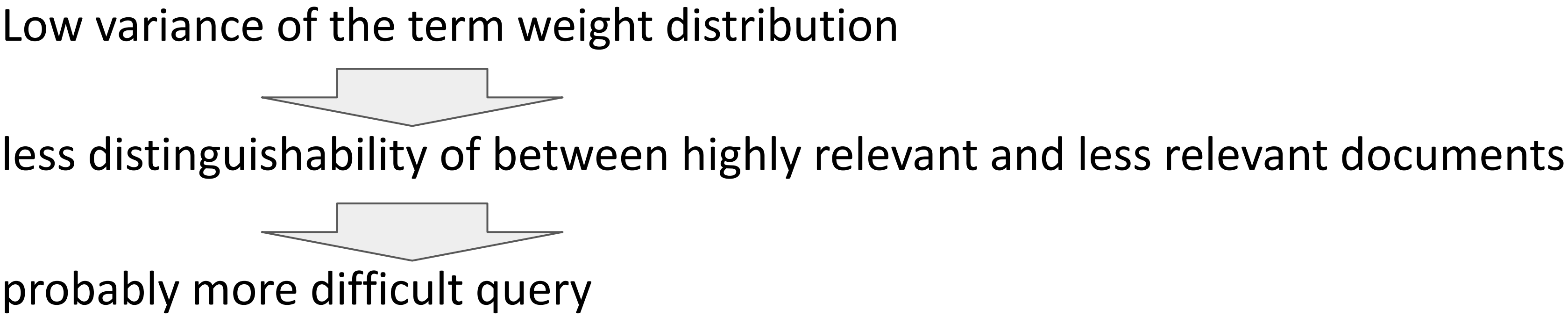
Coherency-based QPP

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 VAR(t): variance of the term weights over the documents containing it in the collection.

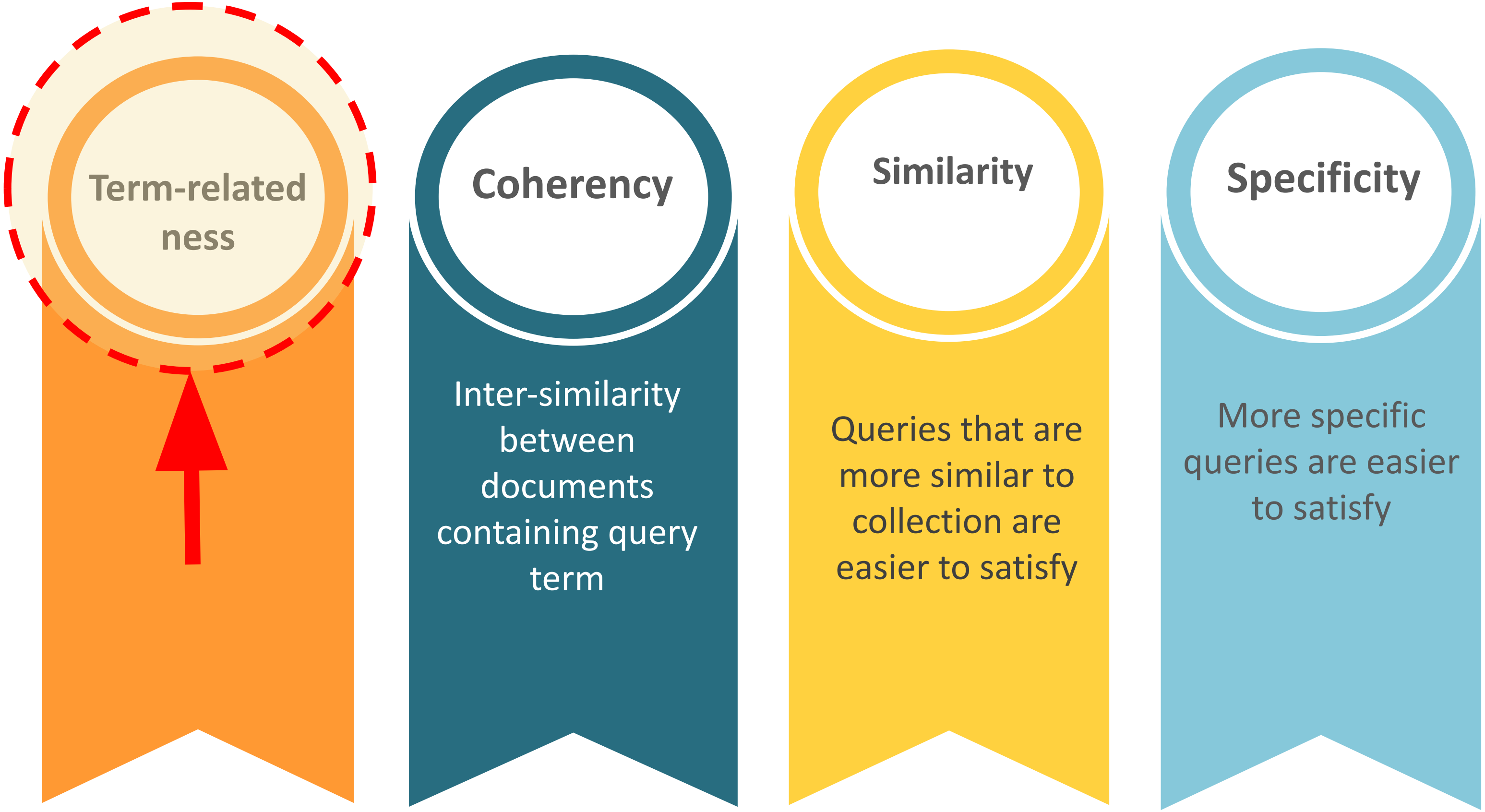


Low variance of the term weight distribution

less distinguishability of between highly relevant and less relevant documents

probably more difficult query

Statistical approaches



Term-relatedness

Coherency

Inter-similarity between documents containing query term

Similarity

Queries that are more similar to collection are easier to satisfy

Specificity

More specific queries are easier to satisfy

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.

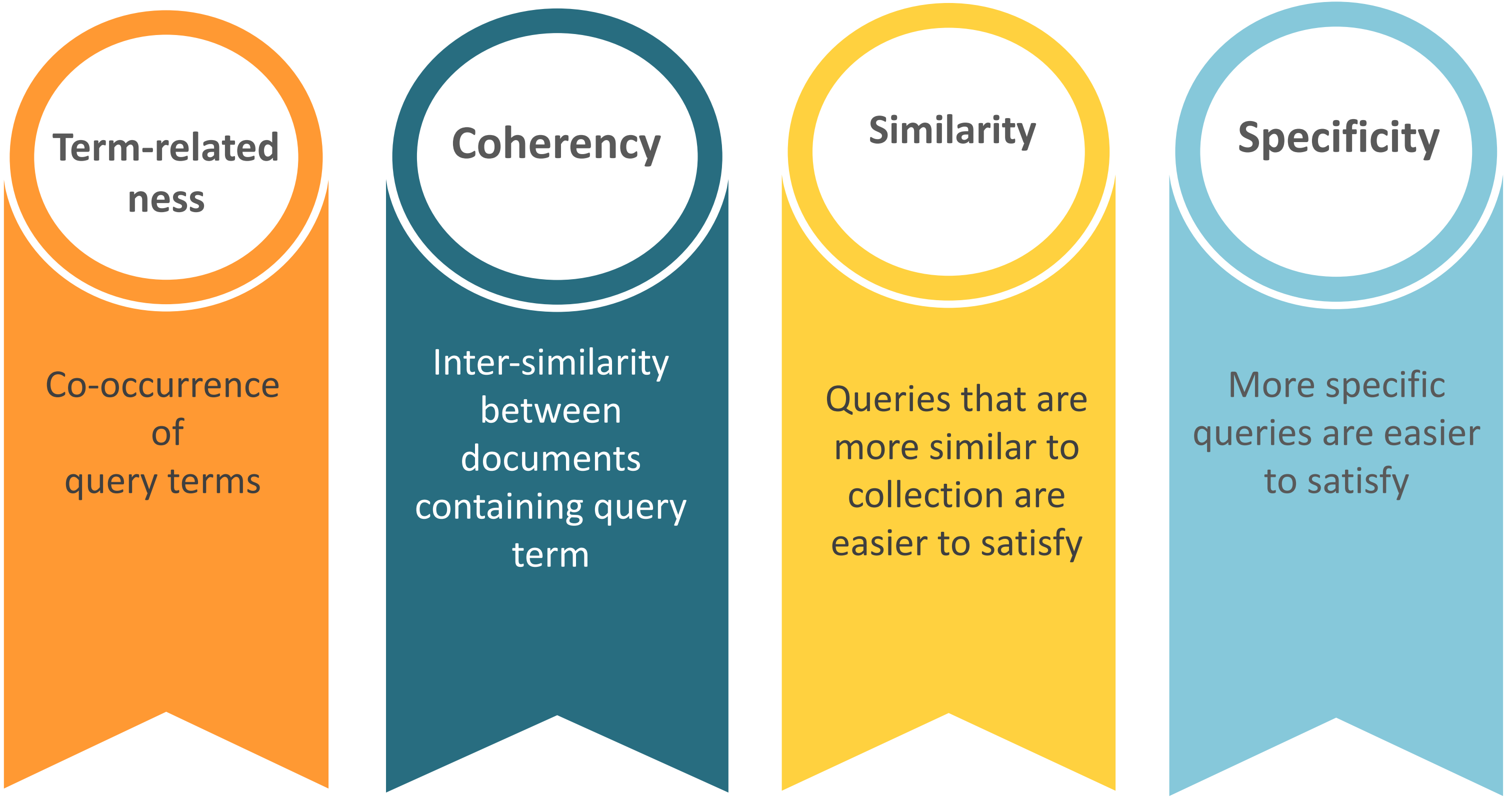
Example: “high blood pressure”

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(t1, t2|D) : the probability of the two terms to co-occur in the corpus.

Statistical approaches



Term-relatedness

Co-occurrence of query terms

Coherency

Inter-similarity between documents containing query term

Similarity

Queries that are more similar to collection are easier to satisfy

Specificity

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

✓

1. Preserve statistical features of terms

Neural Embedding-based metrics

✗

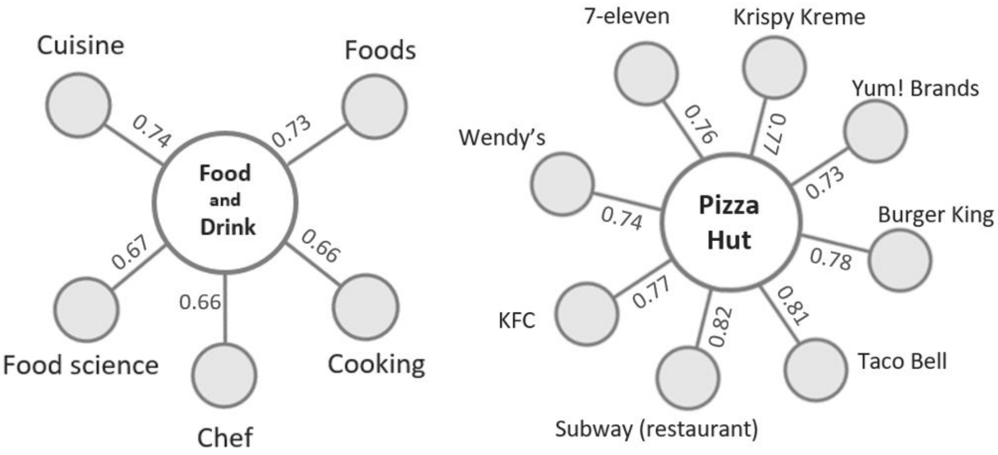
1. Lose the semantic aspects of terms
2. Lose dependency among terms
3. Corpus dependent
4. Complex calculation during index time

Neural-embedding based QPP

ε-neighborhood: Selected local neighborhood surrounding an embedding vector of term t_i, by retrieving a set of highly similar terms to $t_{i.}$

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

Degree of similarity of most similar term to $t_{i.}$



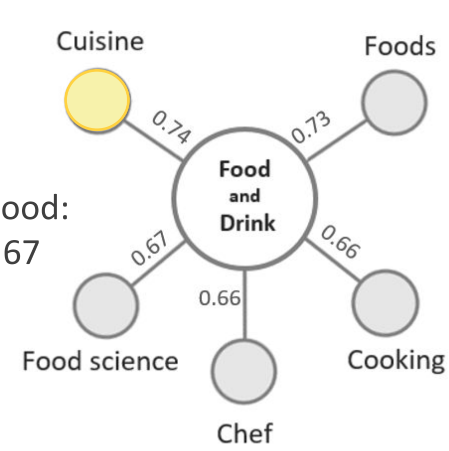
Neural-embedding based QPP

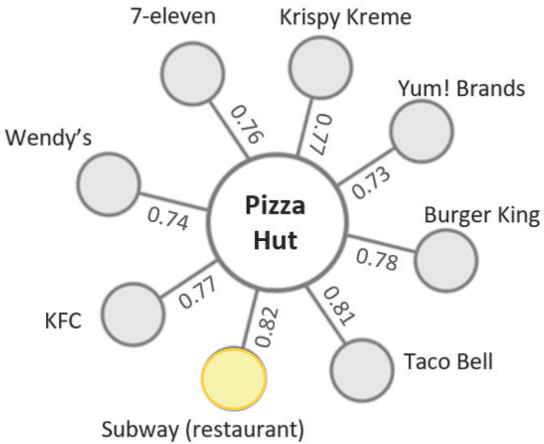
ε-neighborhood: Selected local neighborhood surrounding an embedding vector of term t_i, by retrieving a set of highly similar terms to t_i.

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

Degree of similarity of most similar term to t_i.

0.9-neighborhood:
0.9* 0.74=0.67

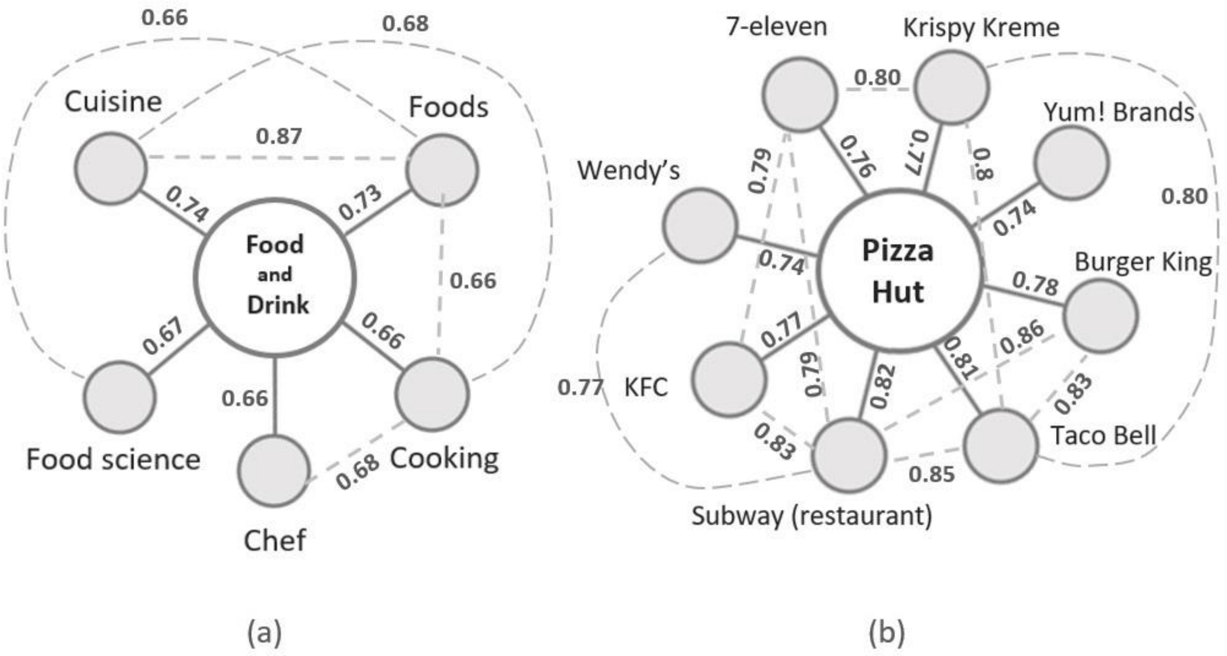




0.9-neighborhood:
0.9*0.82 =0.74

Neural-embedding based QPP

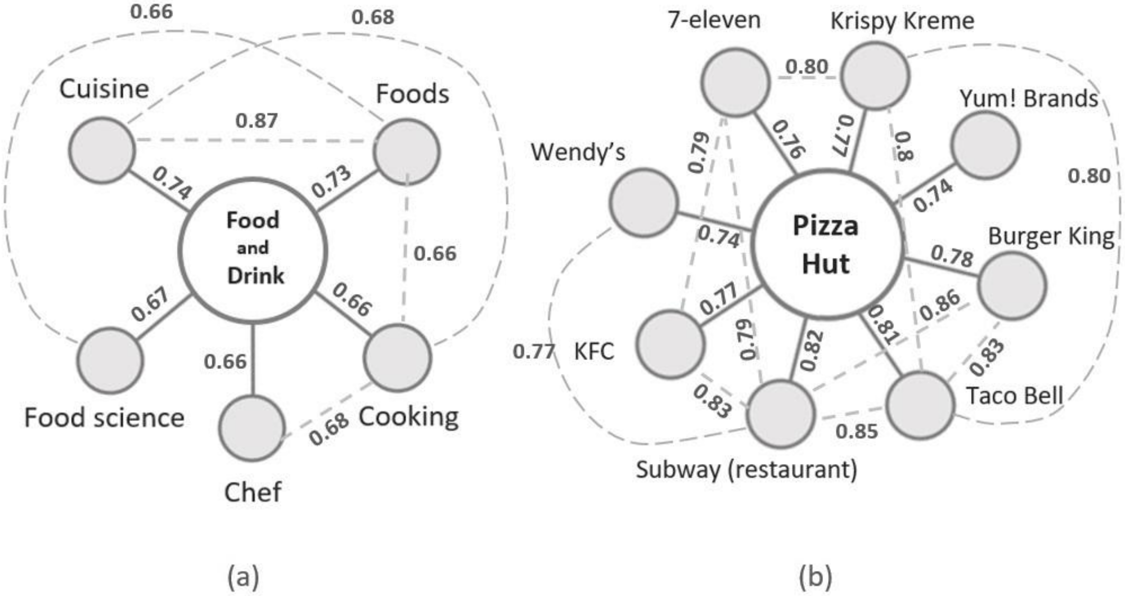
Ego network: t_i is the ego node and is connected directly to other terms only if the degree of similarity between the ego and its neighbors is above a given threshold.



Neural-embedding based QPP

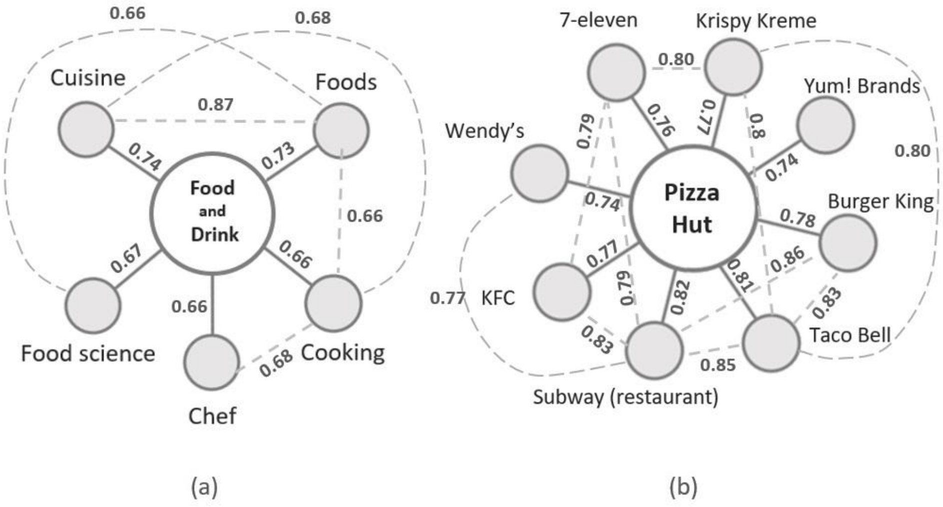
Intuition

- ✔ 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 ⬆



Neural-embedding based QPP

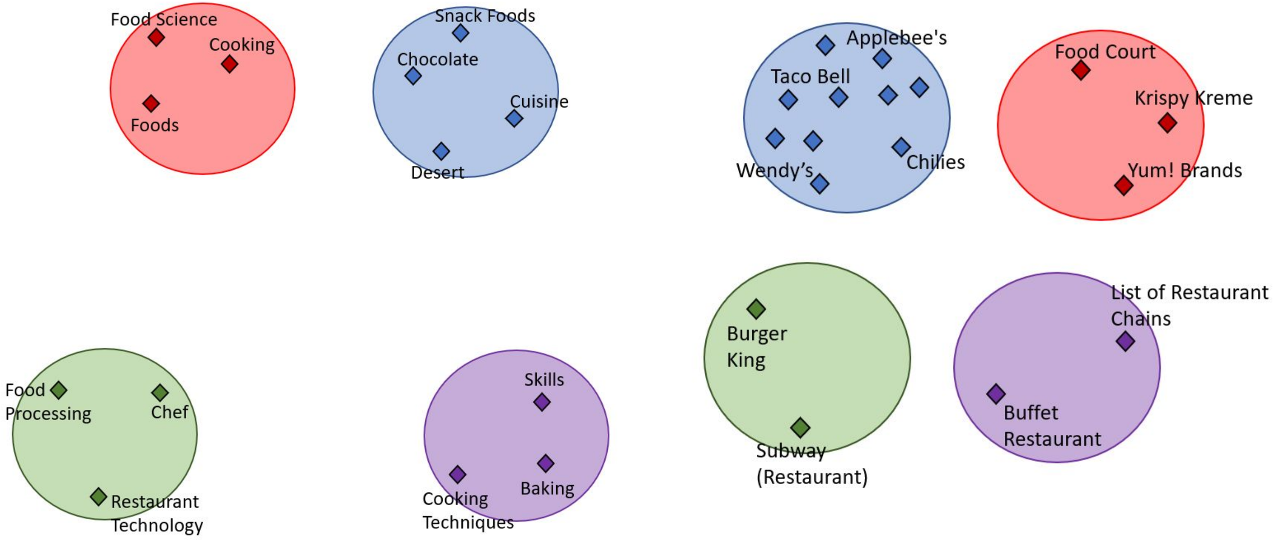
- Neighborhood size (NS)
- Weighted Degree centrality (WDC)
- Median Absolute Deviation (MAD)
- Neighborhood Variance (NV)
- Most Similar Neighbor (MSN)
- Neighborhood Vector similarity (NVS)



Neural-embedding based QPP

Centroid Network steps:

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



Food and Drink

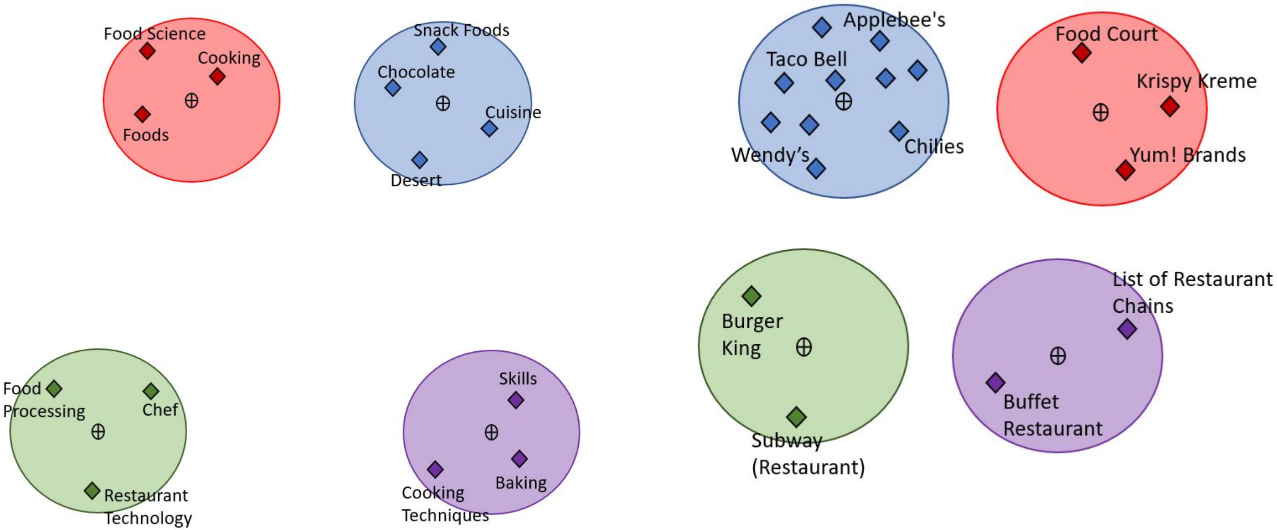
Pizza Hut

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



Food and Drink

Pizza Hut

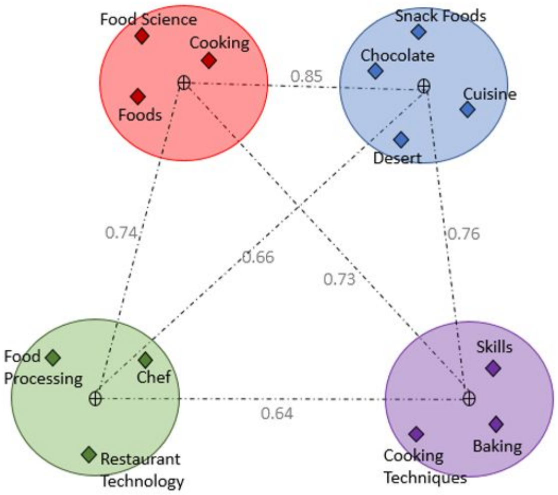
Neural-embedding based QPP

Centroid Network steps:

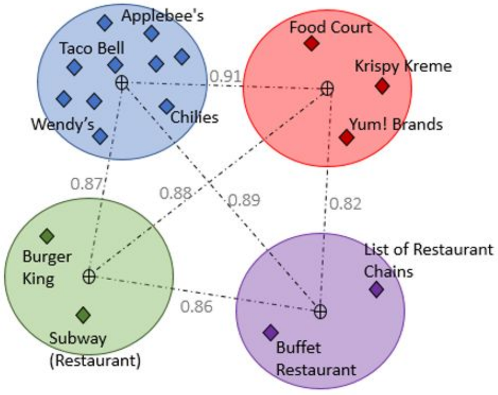
01 We applied K-means clustering algorithm to find K cluster for term t_i $C_{t_i}^1, \dots, C_{t_i}^K$

02 We find the centroid of each cluster

03 We make a weighted graph by connecting all the centroid



Food and Drink

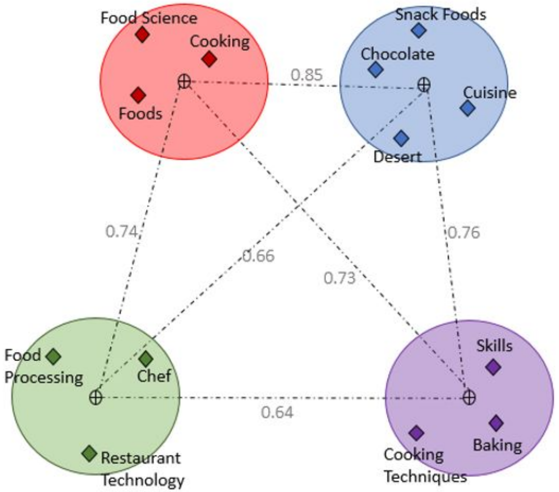


Pizza Hut

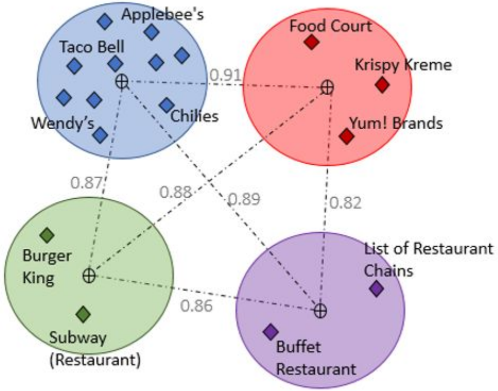
Neural-embedding based QPP

On Centroid Network :

- ❑ Edge Weight Avg_centroid (EWAc)
- ❑ Edge Weight Max_centroid (EWXc)
- ❑ Cluster Elements Variance (CEV)



Food and Drink



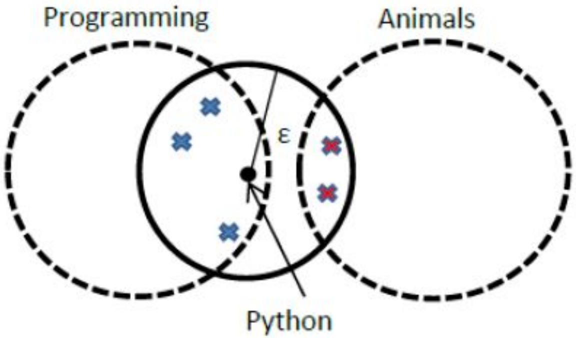
Pizza Hut

Neural-embedding based QPP - Clarity

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

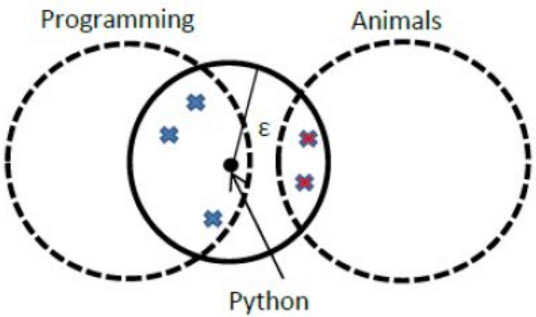
Ambiguous queries

- ○ 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.

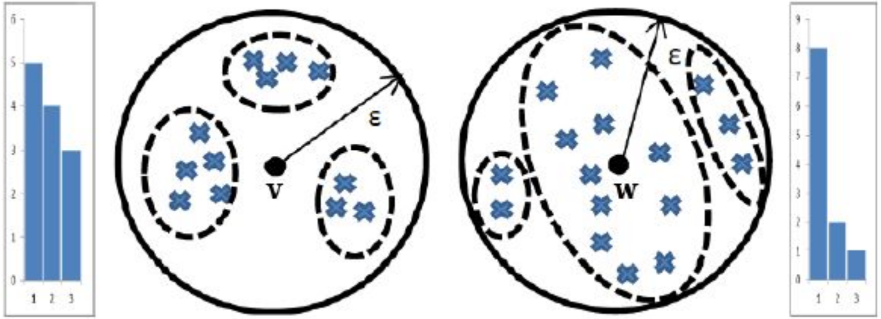
Neural-embedding based QPP

Ɛ-neighbourhood of query terms:

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

Ambiguous term

unambiguous term



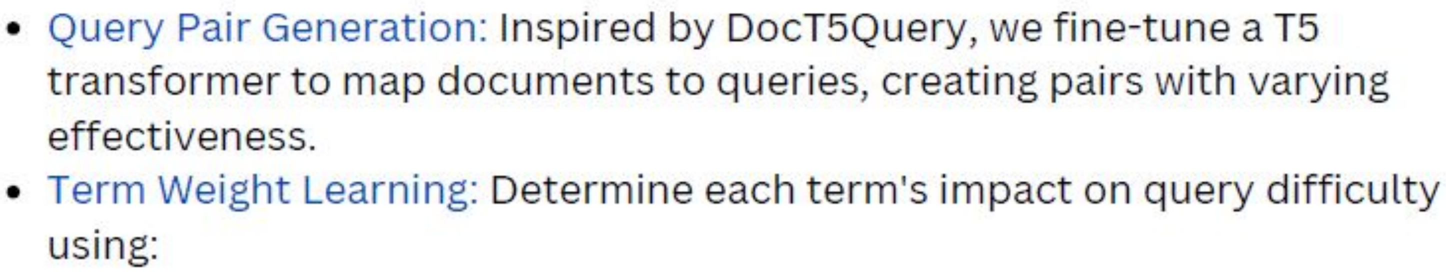
Illustrative diagram of the neighborhood of an ambiguous word with multiple senses

- ➢ 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

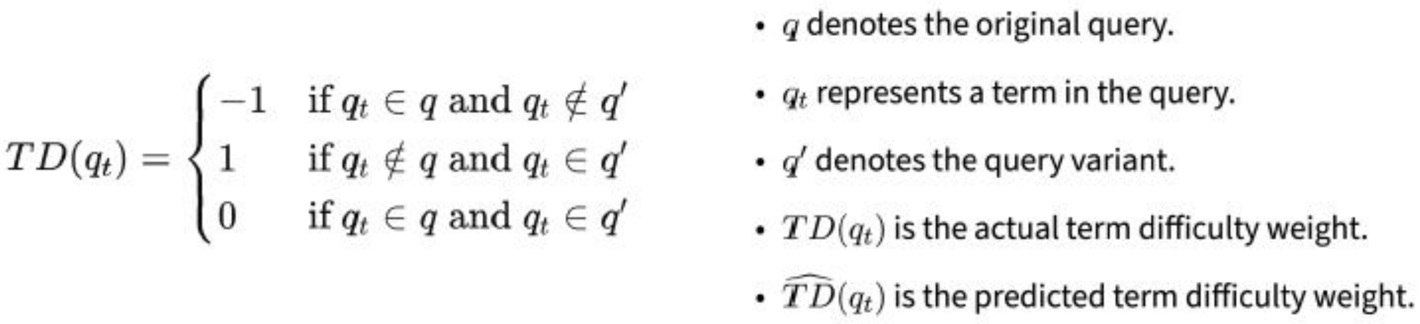
Contextualized-embedding based QPP

Idea: Learning the performance from different query variants

Where to get the query variants?



- Query Pair Generation: Inspired by DocT5Query, we fine-tune a T5 transformer to map documents to queries, creating pairs with varying effectiveness.
- Term Weight Learning: Determine each term's impact on query difficulty using:



$$TD(q_t) = \begin{cases} -1 & \text{if } q_t \in q \text{ and } q_t \notin q' \\ 1 & \text{if } q_t \notin q \text{ and } q_t \in q' \\ 0 & \text{if } q_t \in q \text{ and } q_t \in q' \end{cases}$$

- q denotes the original query.
- q_t represents a term in the query.
- q' denotes the query variant.
- $TD(q_t)$ is the actual term difficulty weight.
- $\widehat{TD}(q_t)$ is the predicted term difficulty weight.

Contextualized-embedding based QPP

Idea: Learning the performance from different query variants

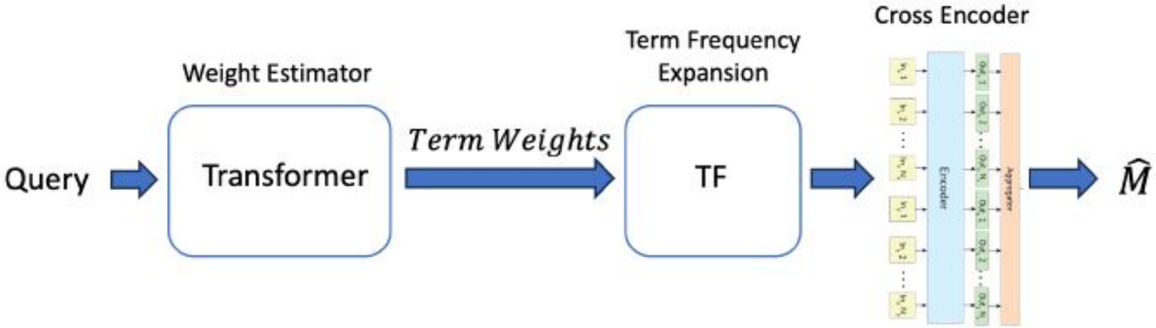
- **Identifying Term Impact:** Determining which query terms impact query performance positively or negatively.

- **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

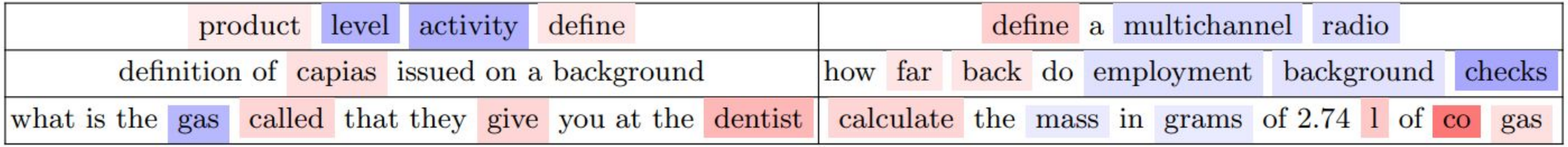
Approach:

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

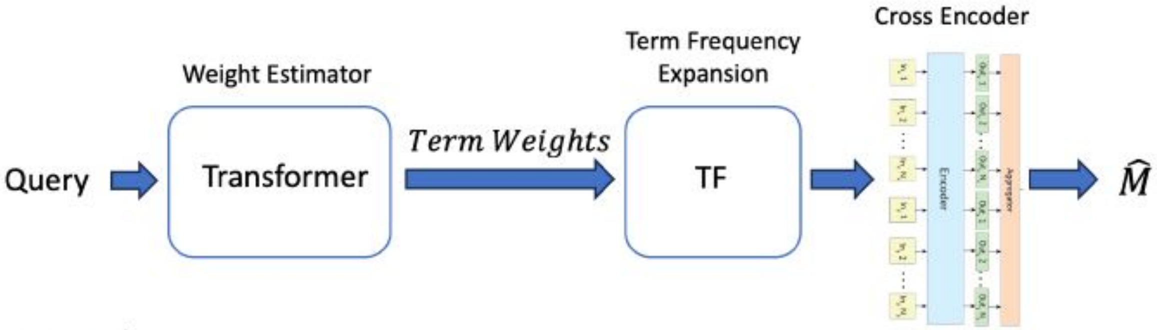
Contextual Adaptation of Term Difficulty:



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

Contextualized-embedding based QPP

Approach:



Example:

Query : how far back do employment background checks

Term Weights : {0.00 , -0.10 , -0.10 , 0.00 , 0.12, 0.11 , 0.34}

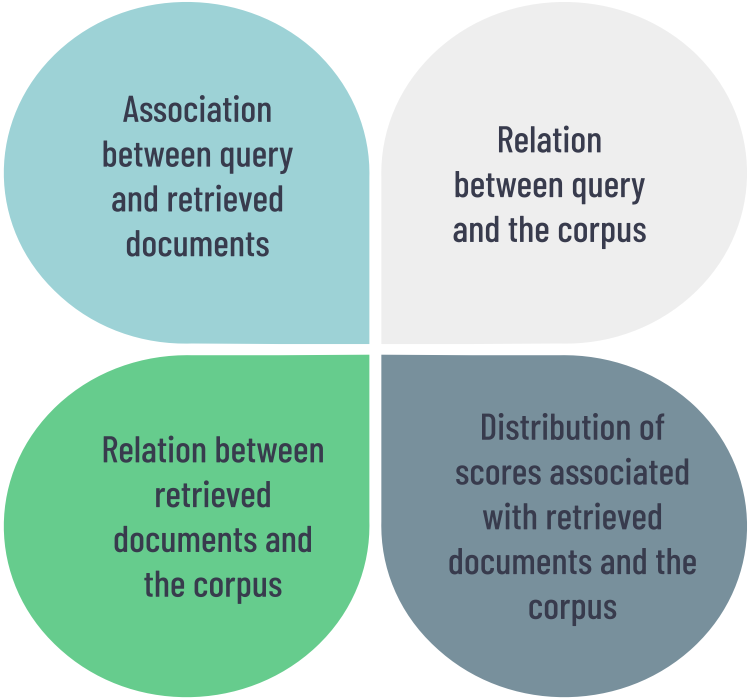
$$\phi^+(q) = \{\text{how far back do } \overbrace{\text{employment employment ... employment}}^{\propto TF\ (0.12)}\ \underbrace{\text{background background ... background}}_{\propto TF\ (0.11)}\ \underbrace{\text{checks checks ... } checks}_{\propto TF\ (0.34)}\}$$

$$\phi^-(q) = \{\text{how } \underbrace{\text{far far ... far}}_{\propto TF\ (0.10)}\ \underbrace{\text{back back ... back}}_{\propto TF\ (0.10)} \text{ do employment background checks}\}$$

Q & A

Post-retrieval QPP

Post retrieval QPP



Association between query and retrieved documents

Relation between query and the corpus

Relation between retrieved documents and the corpus

Distribution of scores associated with retrieved documents and the corpus

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.

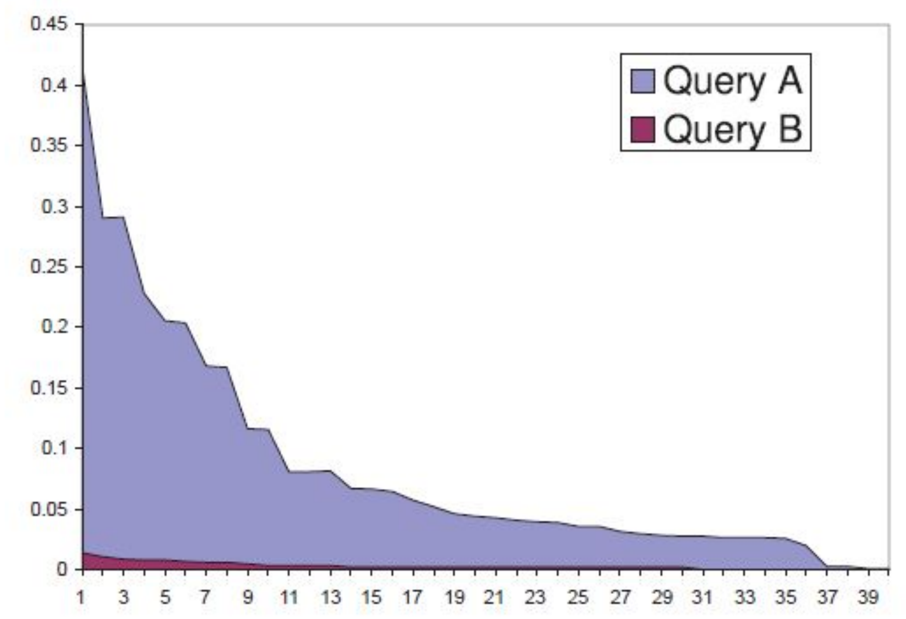
$$Clarity(q) = KL_{div}(Pr(\cdot|D_q)||Pr(\cdot|D)) = \sum_{t \in V(D)} Pr(t|D_q) \log \frac{Pr(t|D_q)}{Pr(t|D)}$$

- **Potential downside**: efficiency

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

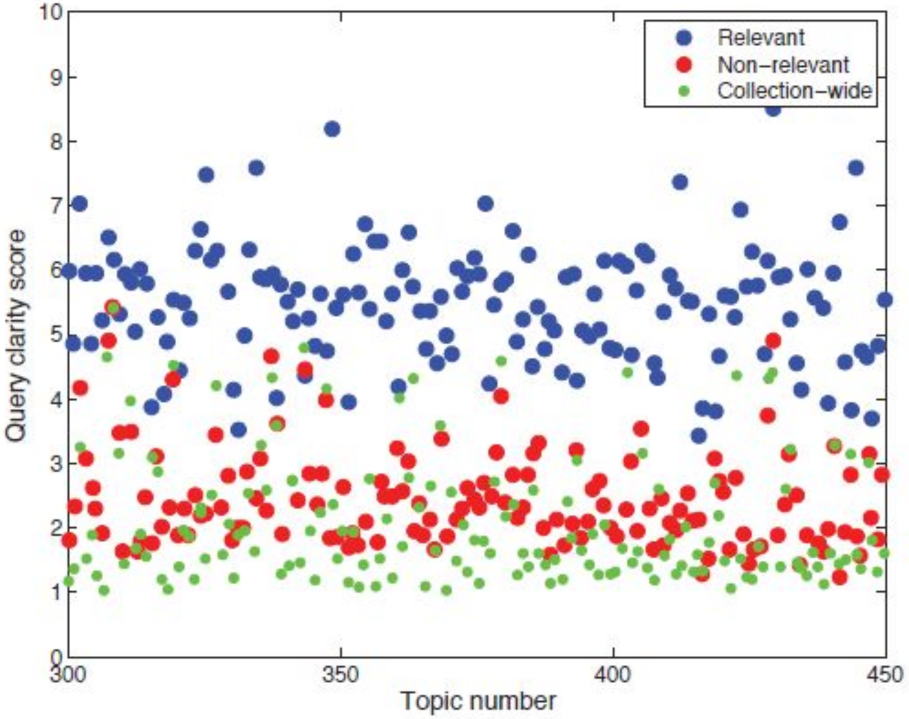
Coherency-based QPP

- ➢ **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.

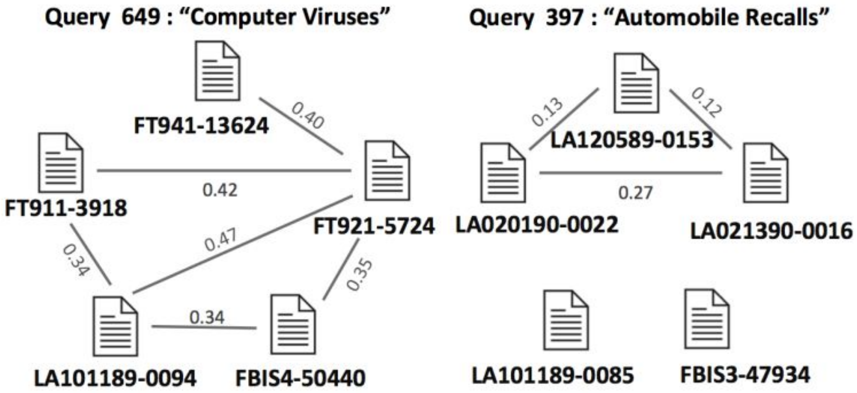
Assumption:

- ➢ 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.

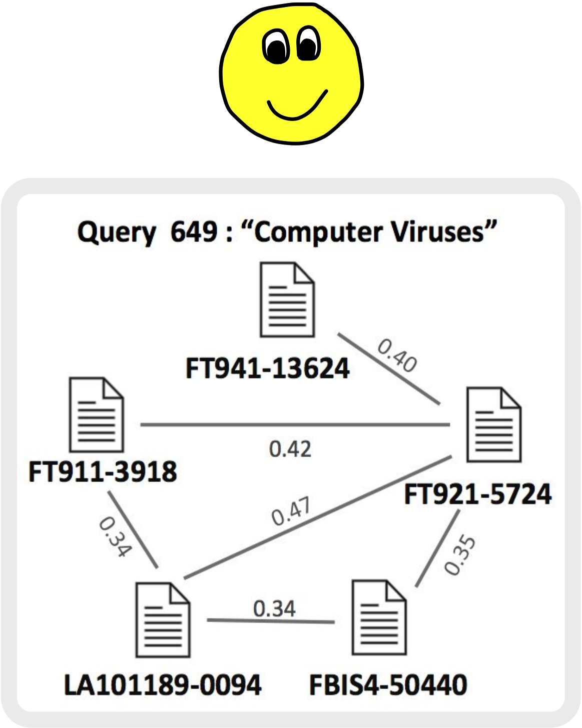


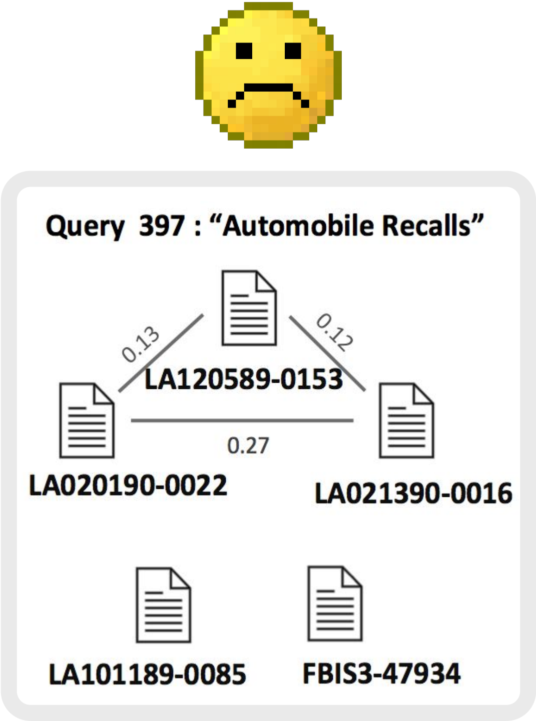
Coherency-based QPP

- **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

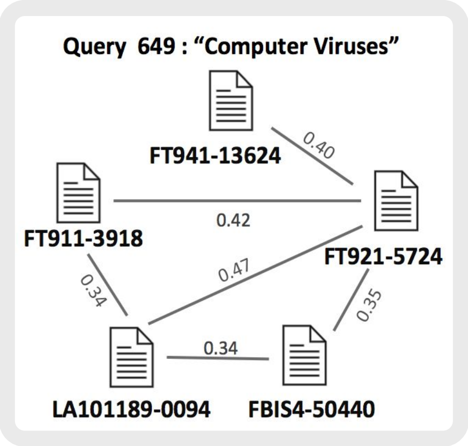




Coherency-based QPP

Higher number of document associations

Denser

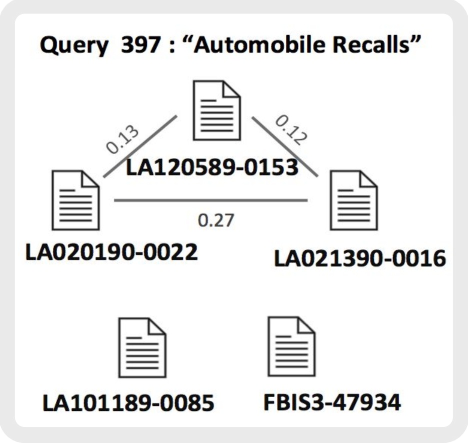


Query 649 : “Computer Viruses”

Lower edge weights

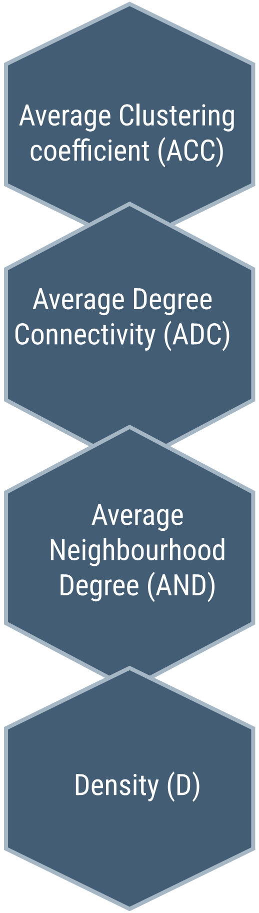
Disconnected

Sparse



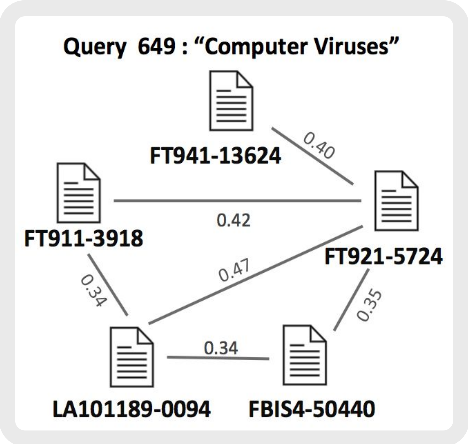
Query 397 : “Automobile Recalls”

Coherency-based QPP


Average Clustering coefficient (ACC)
Average Degree Connectivity (ADC)
Average Neighbourhood Degree (AND)
Density (D)

Higher number of document associations

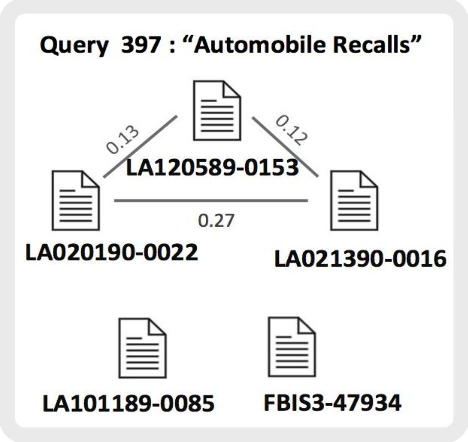
Denser



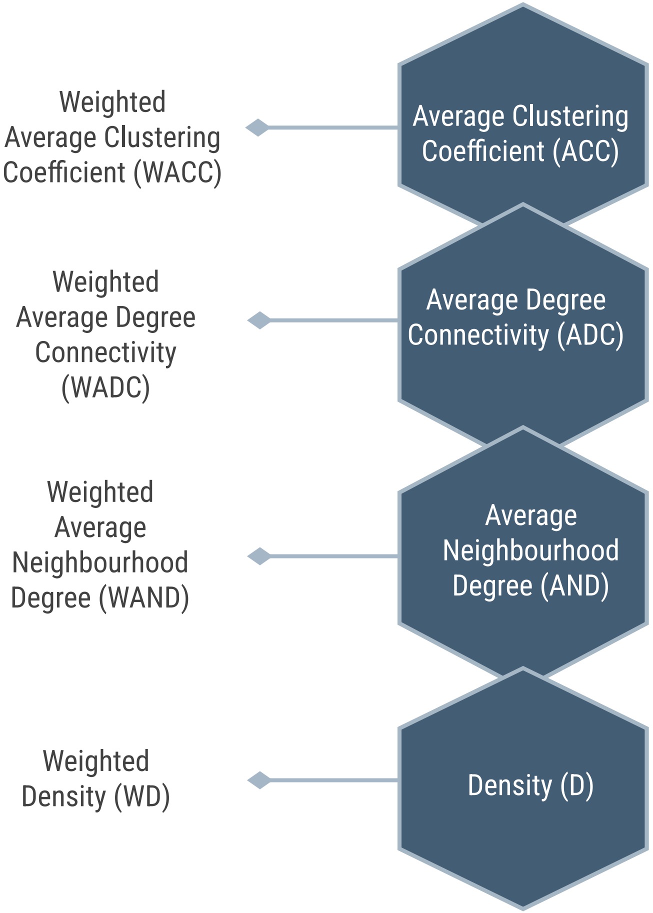
Lower edge weights

Disconnected

Sparse



Coherency-based QPP



Weighted Average Clustering Coefficient (WACC)

Average Clustering Coefficient (ACC)

Weighted Average Degree Connectivity (WADC)

Average Degree Connectivity (ADC)

Weighted Average Neighbourhood Degree (WAND)

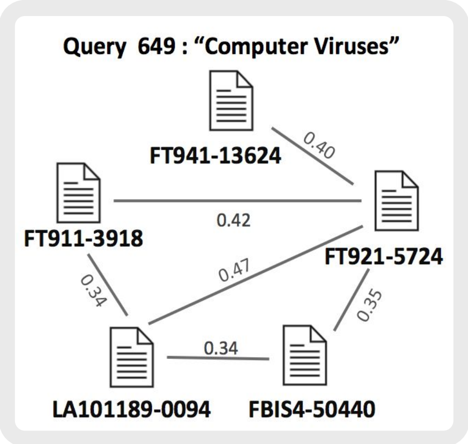
Average Neighbourhood Degree (AND)

Weighted Density (WD)

Density (D)

Higher number of document associations

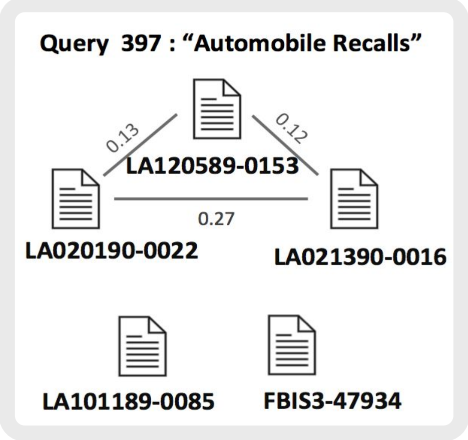
Denser



Lower edge weights

Disconnected

Sparse



Coherency-based QPP

Given:

- ➢ Coh(G): coherence metrics on the document association network (G).
- ➢ QPP(q, D_q, C): predictor of choice for query q and the list of top-k retrieved documents from corpus C.

Interpolated score:

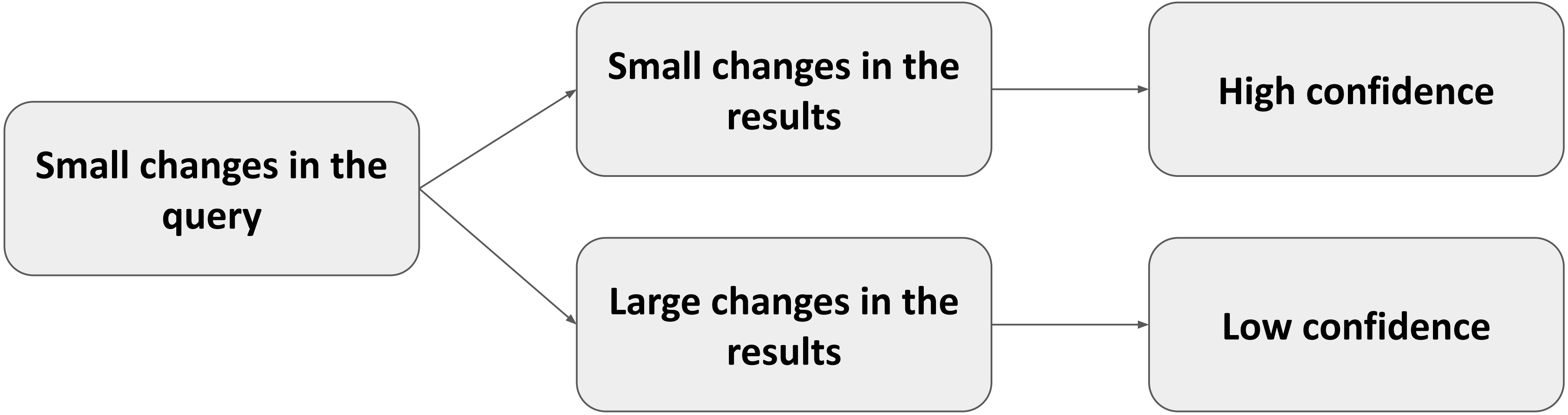
$$\mu(q, D_q, C) = \lambda.QPP(q, D_q, C) + (1 - \lambda)\ Coh(G)$$

Robustness-based QPP

Query perturbation

Idea:

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



Small changes in the query

Small changes in the results

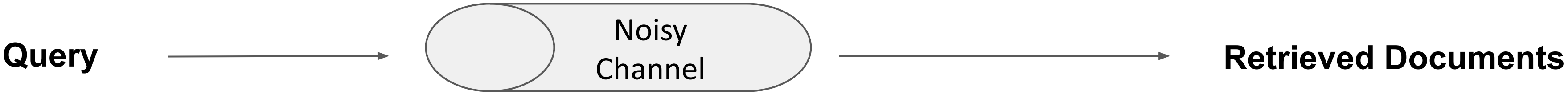
Large changes in the results

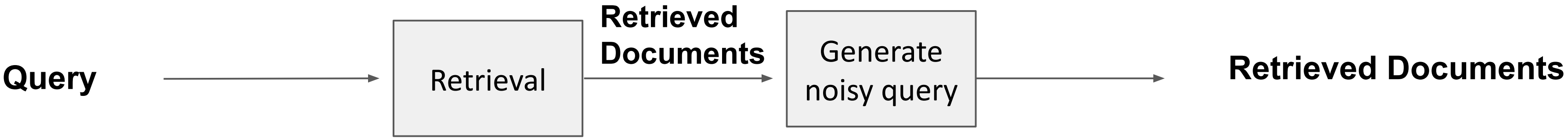
High confidence

Low confidence

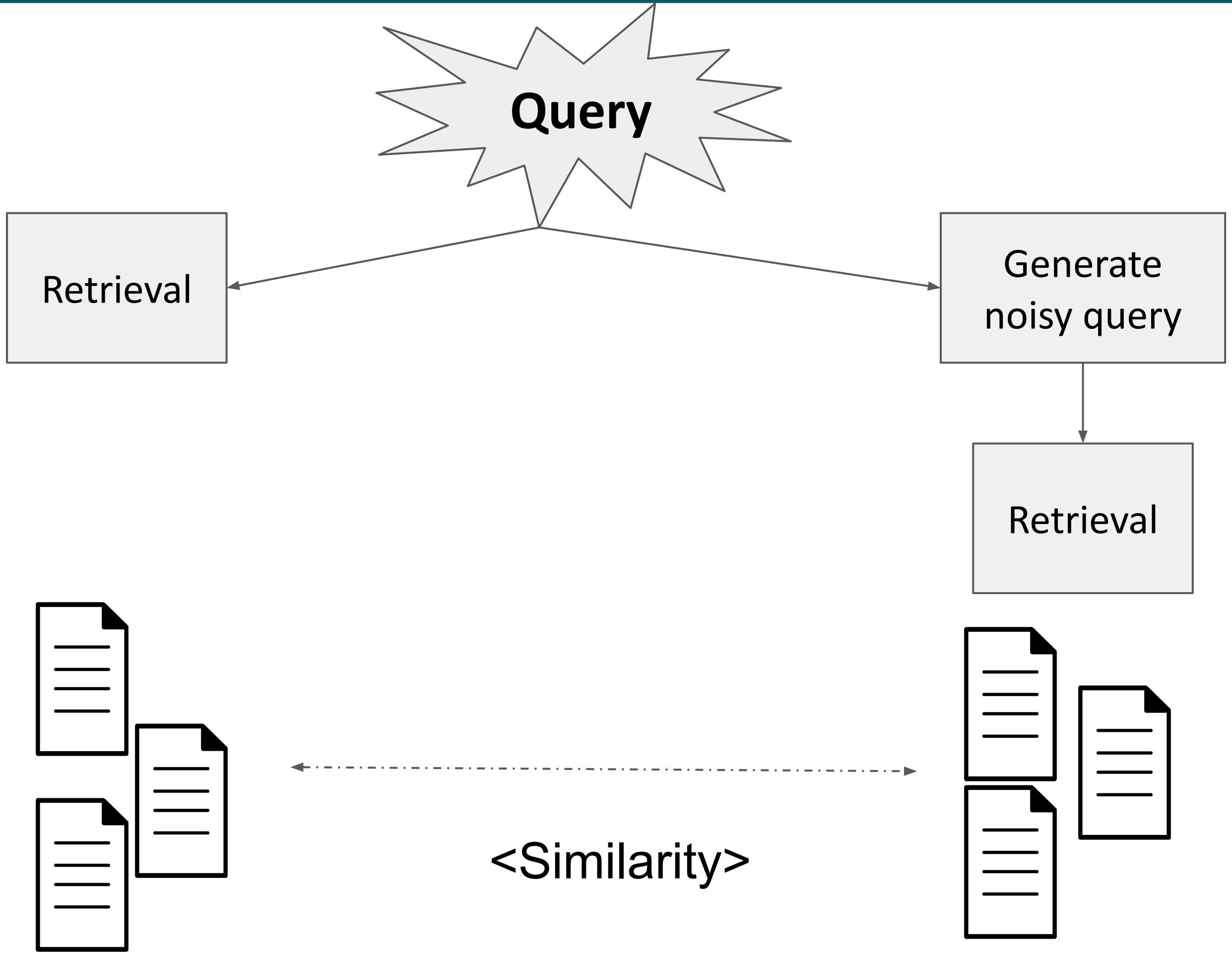
Query perturbation

Query Feedback:Models retrieval as a communication channel problem.





Query perturbation



Query perturbation

Idea: Perturbation with sub-queries.

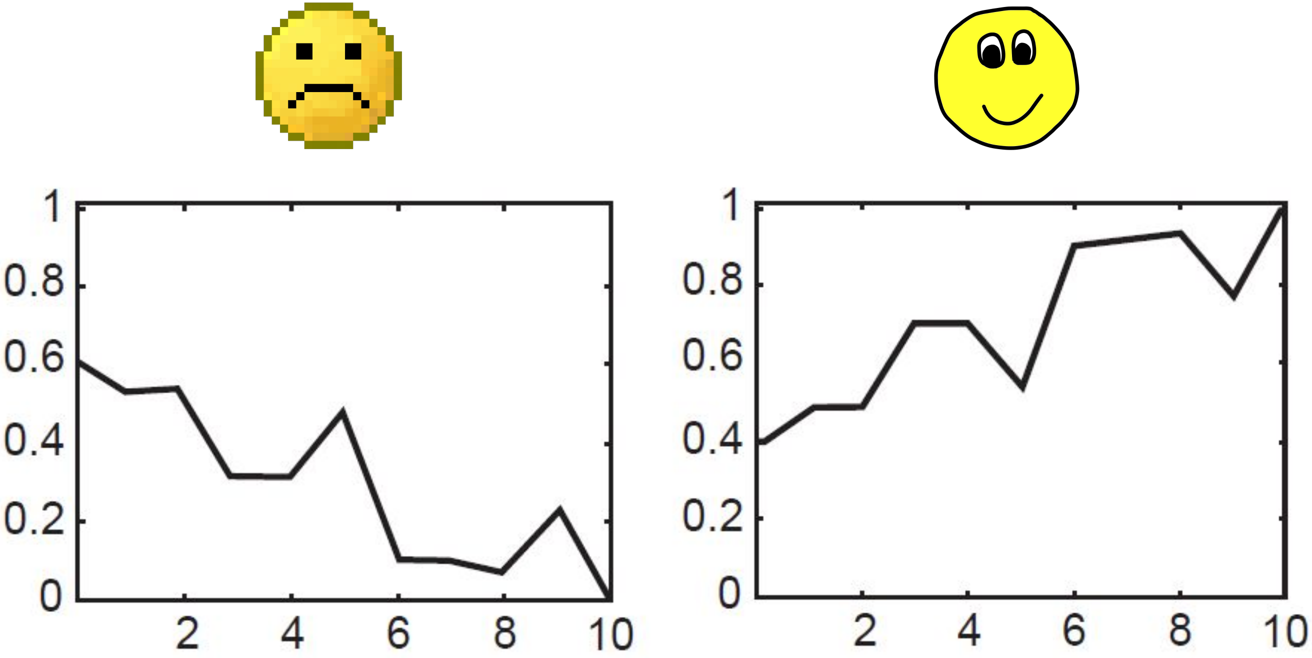
Approach:

- ➢ 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



Query perturbation

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

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

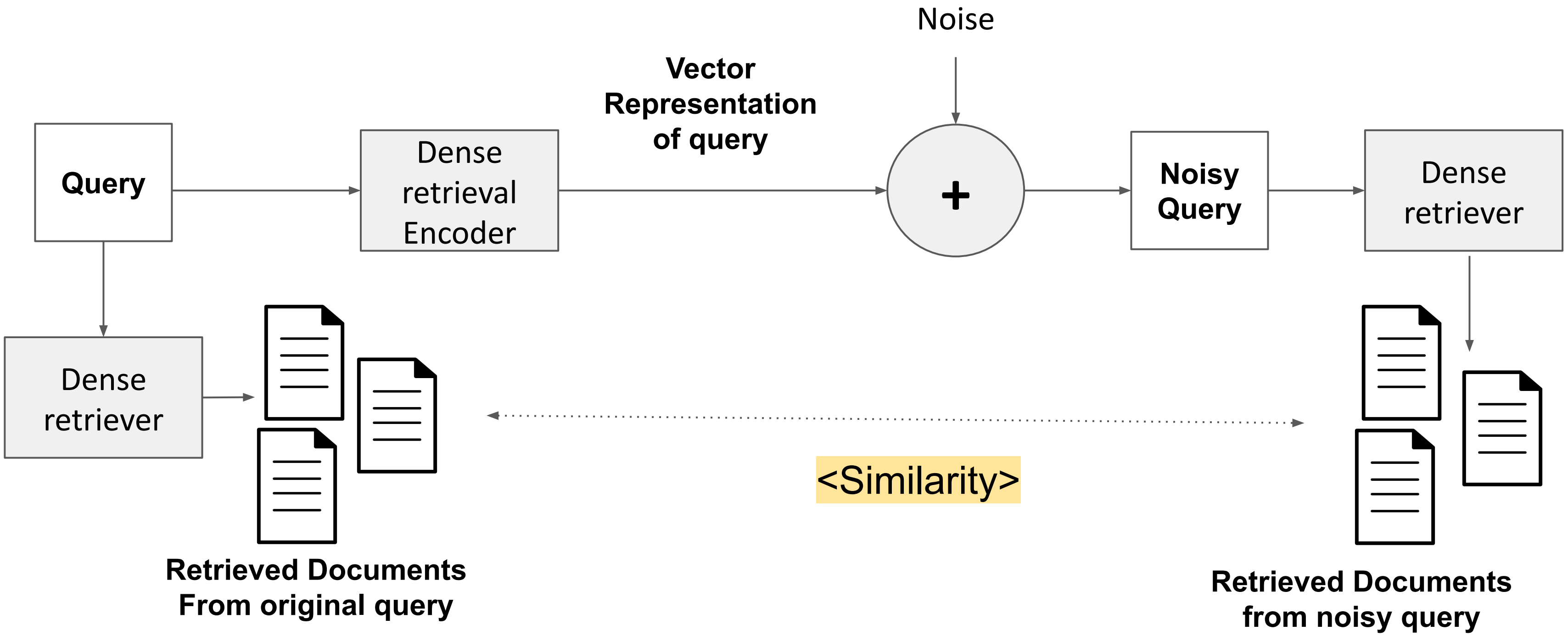
Generate query perturbations for measuring query robustness

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.


Noise
Vector Representation of query
Query
Dense retrieval Encoder
+
Noisy Query
Dense retriever
Dense retriever
<Similarity>
Retrieved Documents From original query
Retrieved Documents from noisy 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.

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

λ(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?

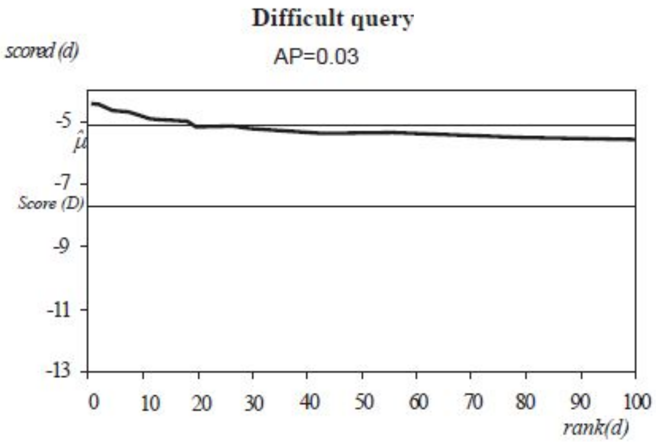
Higher variance in scores → easier distinguishability of items

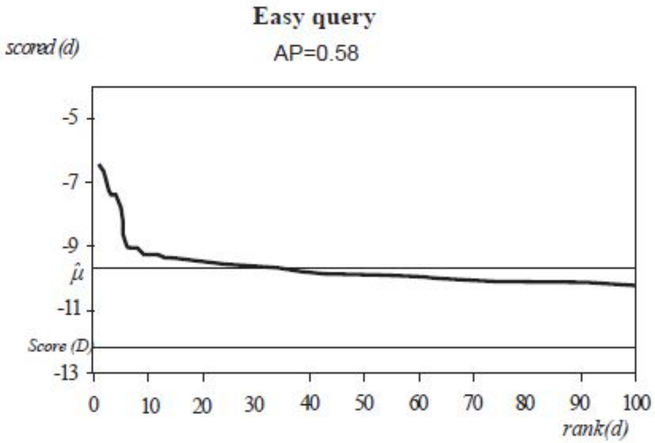
$$NQC(q) = \frac{\sqrt{\frac{1}{k}\sum_{d \in D_q^k}(Score(d) - \mu)^2}}{|Score(D)|}.$$

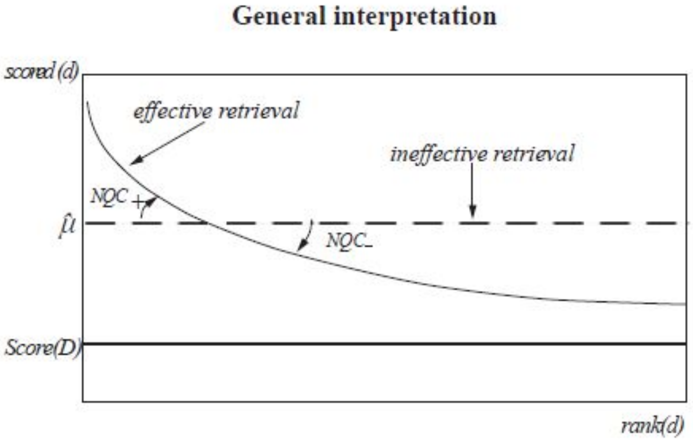
Measuring standard deviation of retrieval scores in the top-retrieved document

normalizing if by the whole collection score

Score-based QPP - NQC



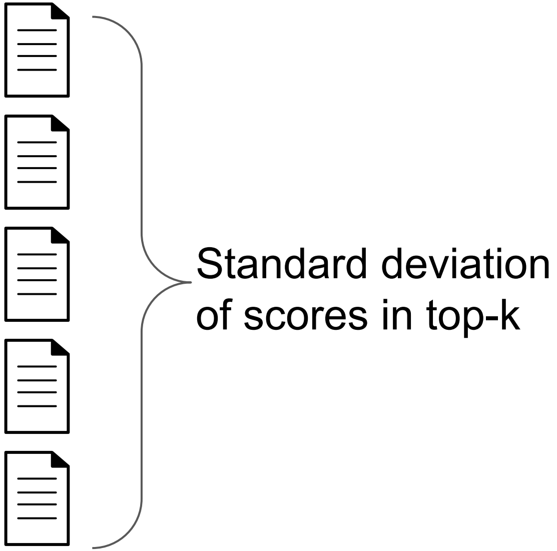


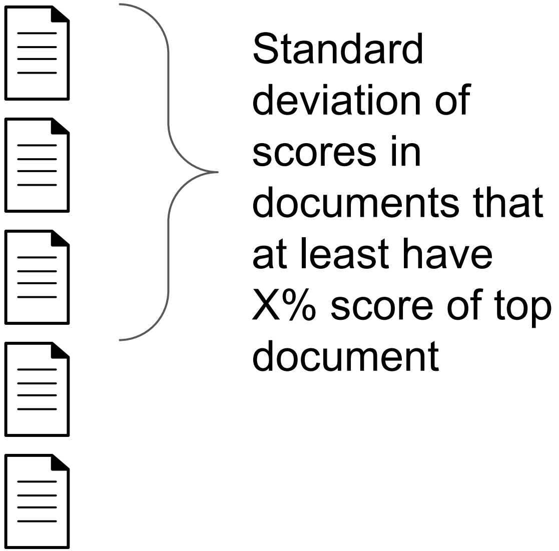


Score-based QPP - Dynamic Cut off

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.





Standard deviation of scores in documents that at least have X% score of top document

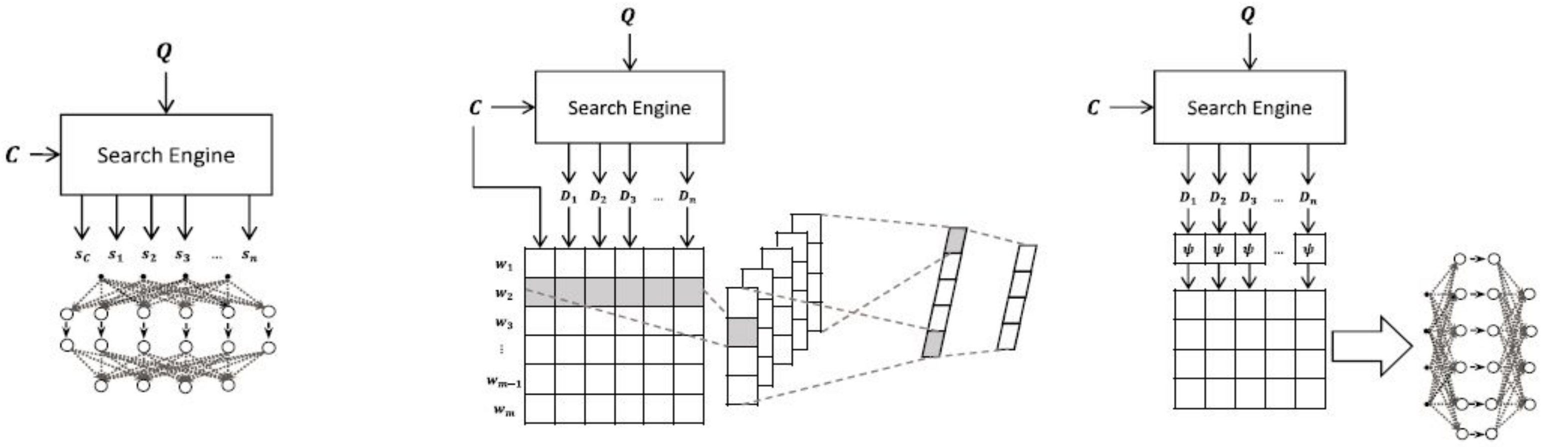
Score-based QPP - Robust Standard Deviation Estimation

Idea: Dynamic selection of documents for QPP by modeling user behaviour

- Previous works directly estimate standard deviation from the original result list.
- Mimics a "random user" browsing and selecting documents from search results for QPP.
- Rank-biased selection: Higher-ranked documents have a higher chance of selection.
- Utilizes bootstrap sampling approach for estimating standard deviation.
- Without replacement and round-robin sampling to cover the top documents.

Embedding-based QPP
Post retrieval

Neural-based QPP



(a) Retrieval Scores Analyzer

(b) Term Distribution Analyzer

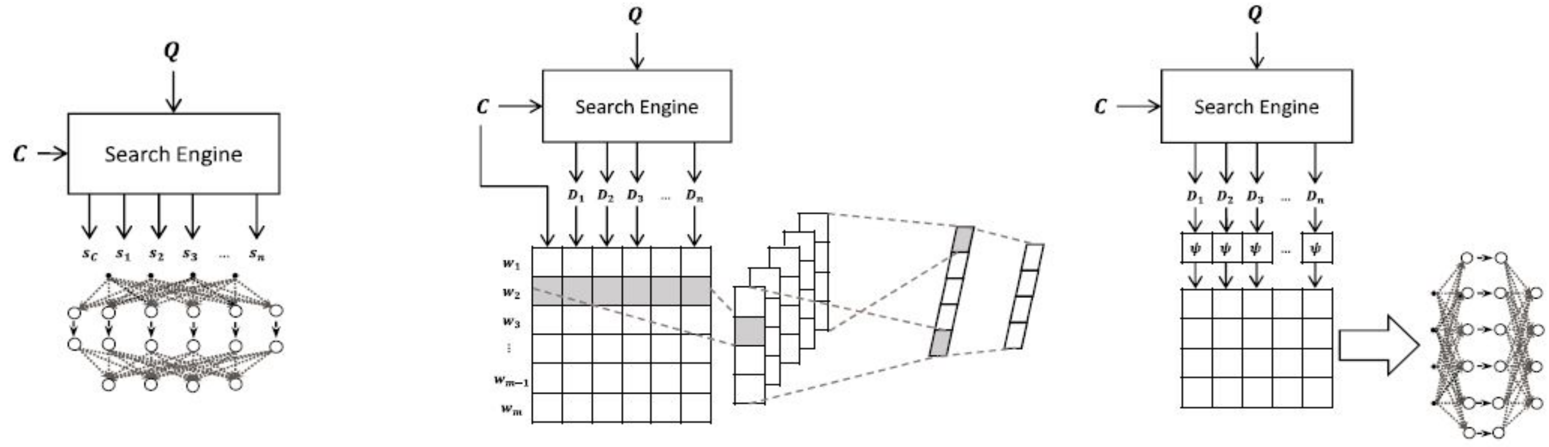
(c) Semantic Analyzer

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

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

Zamani et al. “Neural Query Performance Prediction using Weak Supervision from Multiple Signals”

Neural-based QPP


Q
C
Search Engine
s_C s_1 s_2 s_3 ... s_n
Q
C
Search Engine
D_1 D_2 D_3 ... D_n
w_1
w_2
w_3
⋮
w_{m-1}
w_m
Q
C
Search Engine
D_1 D_2 D_3 ... D_n
ψ ψ ψ ... ψ

(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

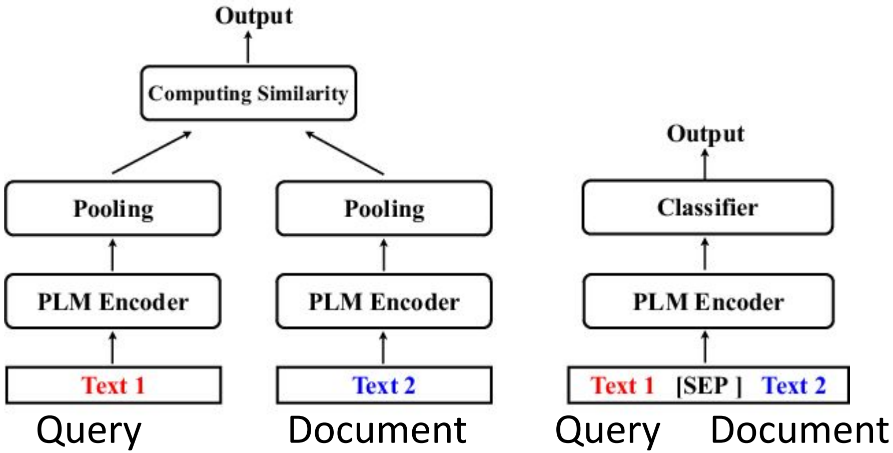
Zamani et al. “Neural Query Performance Prediction using Weak Supervision from Multiple Signals”

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

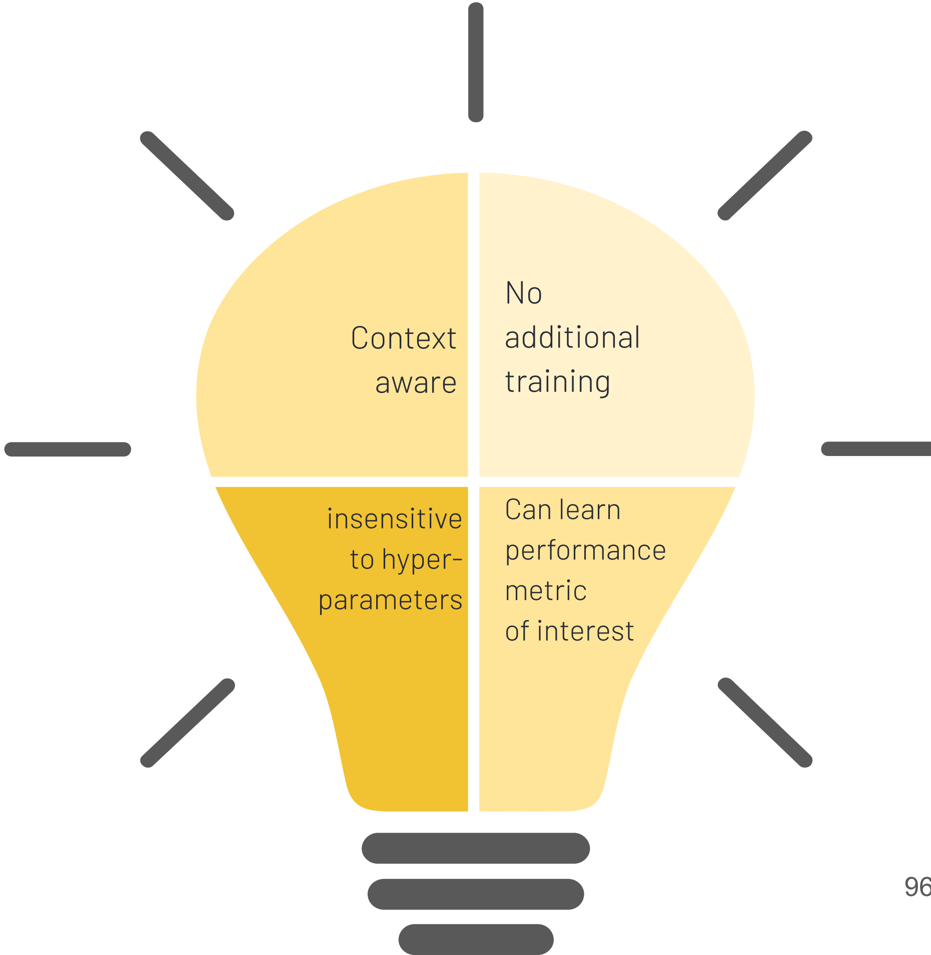
- ➢ Learning the relevance → Learning the performance



(a) Bi-Encoder (b) Cross-Encoder

BERT-QPP

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



Context aware

No additional training

insensitive to hyper-parameters

Can learn performance metric of interest

BERT-QPP

	BERT-QPP$_{bi}$	Bert-QPP$_{cross}$
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 × 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$_{cross}$	2.6
BERT-QPP$_{bi}$	0.7

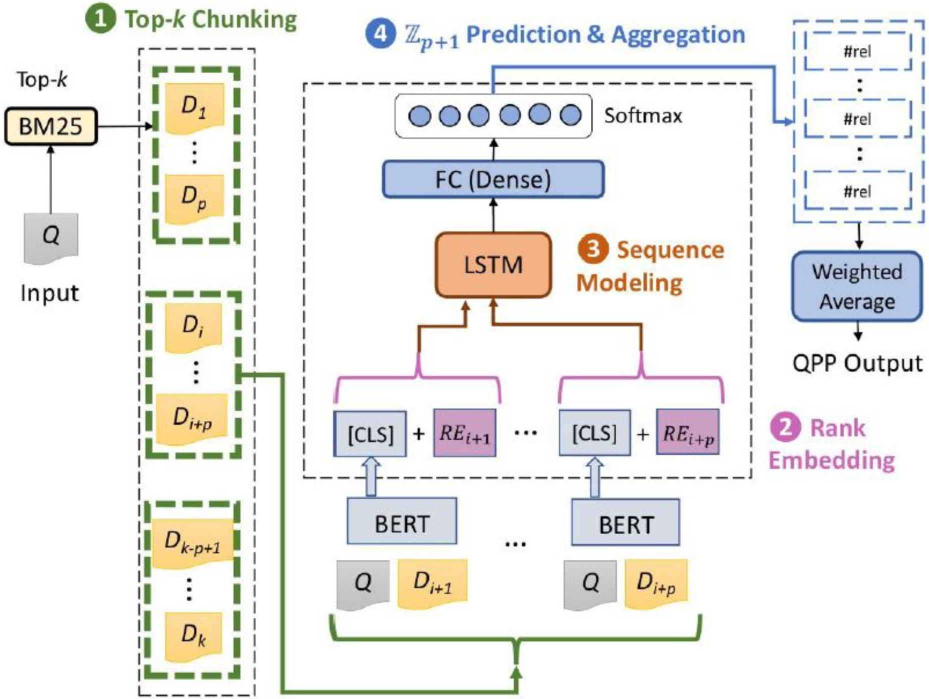
qppBERT-pl

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



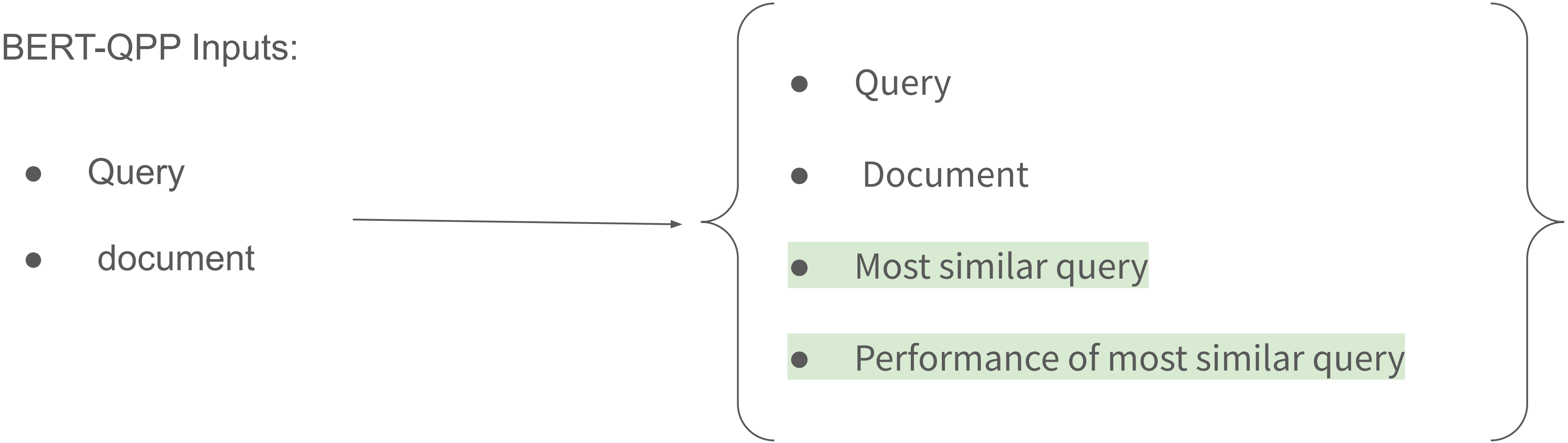
Enriched BERT-QPP

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:

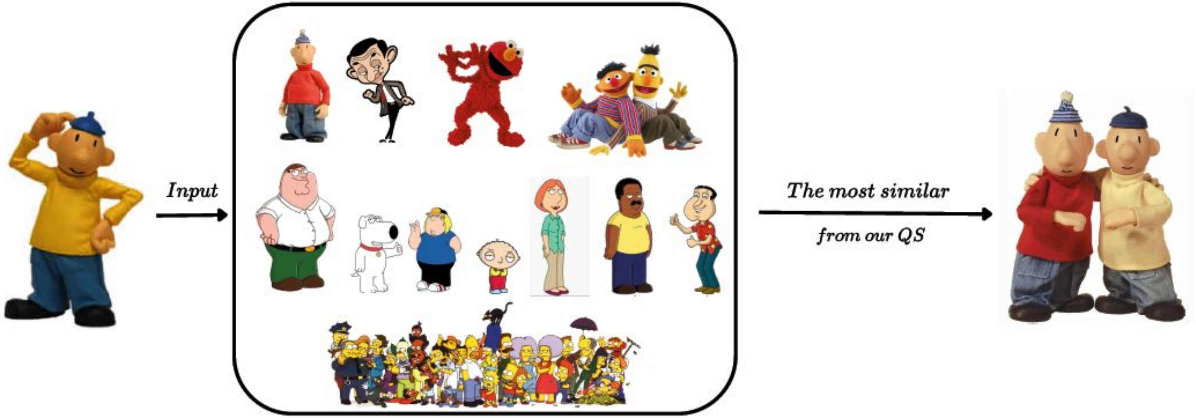


- Query
- document

- Query
- Document
- Most similar query
- Performance of most similar query

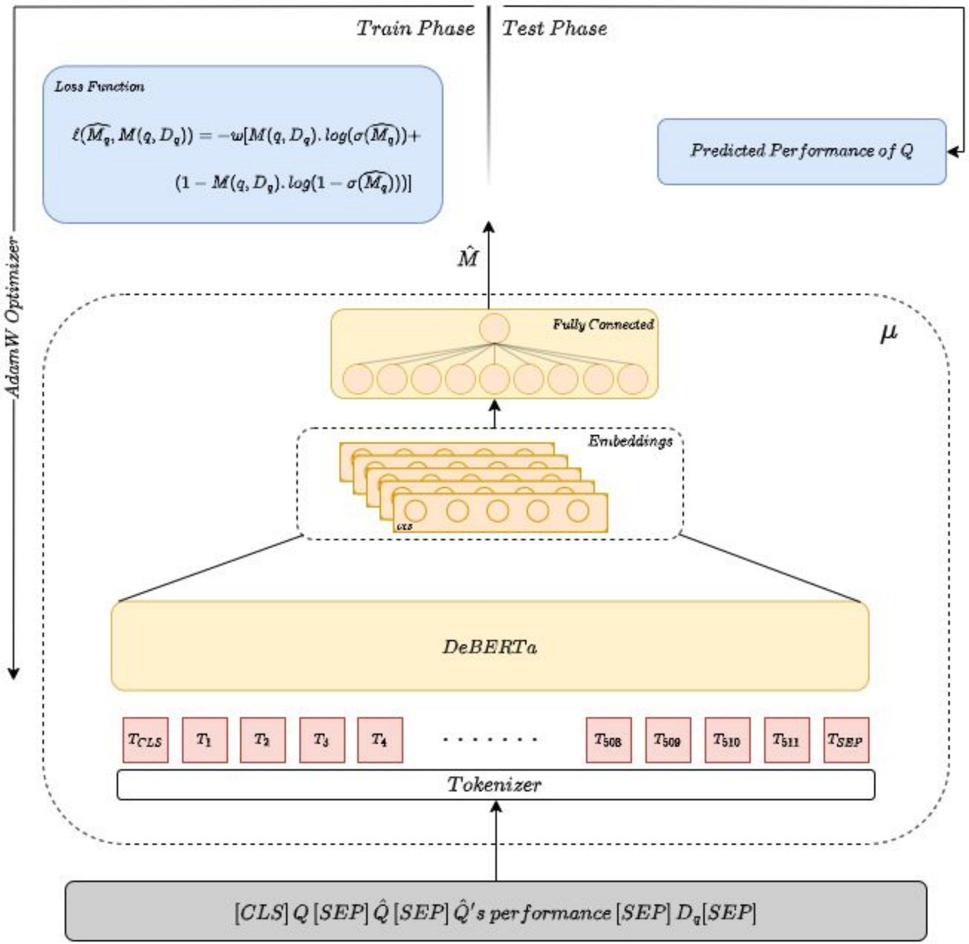
Enriched BERT-QPP

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

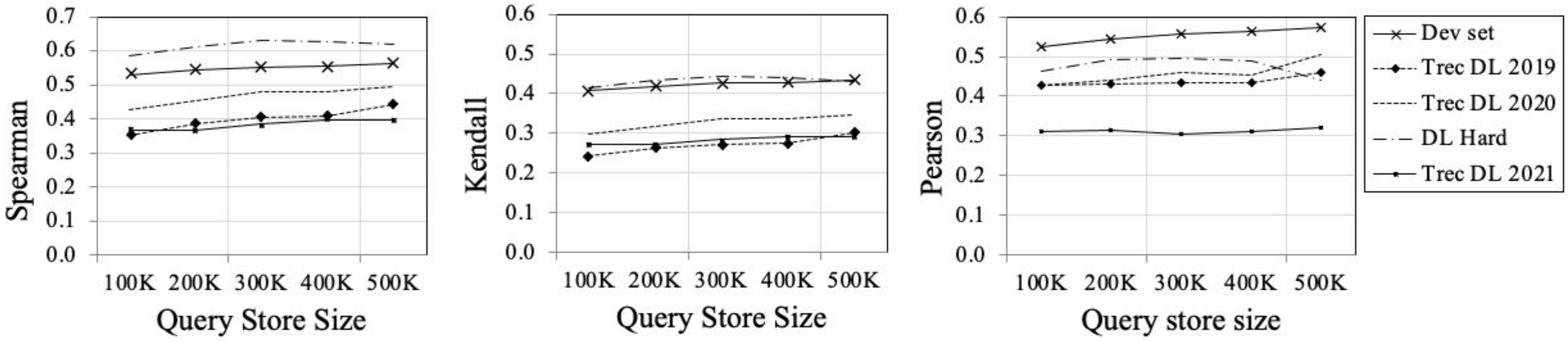
Enriched BERT-QPP



Train Phase
Test Phase
Loss Function
$\ell(\widehat{M_q}, M(q, D_q)) = -w[M(q, D_q).log(\sigma(\widehat{M_q}))+$
$(1 - M(q, D_q).log(1 - \sigma(\widehat{M_q}))]$
Predicted Performance of Q
AdamW Optimizer
$\hat{M}$
μ
Fully Connected
Embeddings
cls
DeBERTa
T_{CLS}
T_1
T_2
T_3
T_4
. . . .
T_{508}
T_{509}
T_{510}
T_{511}
T_{SEP}
Tokenizer
$[CLS]\ Q\ [SEP]\ \hat{Q}\ [SEP]\ \hat{Q}'s\ performance\ [SEP]\ D_q[SEP]$

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:

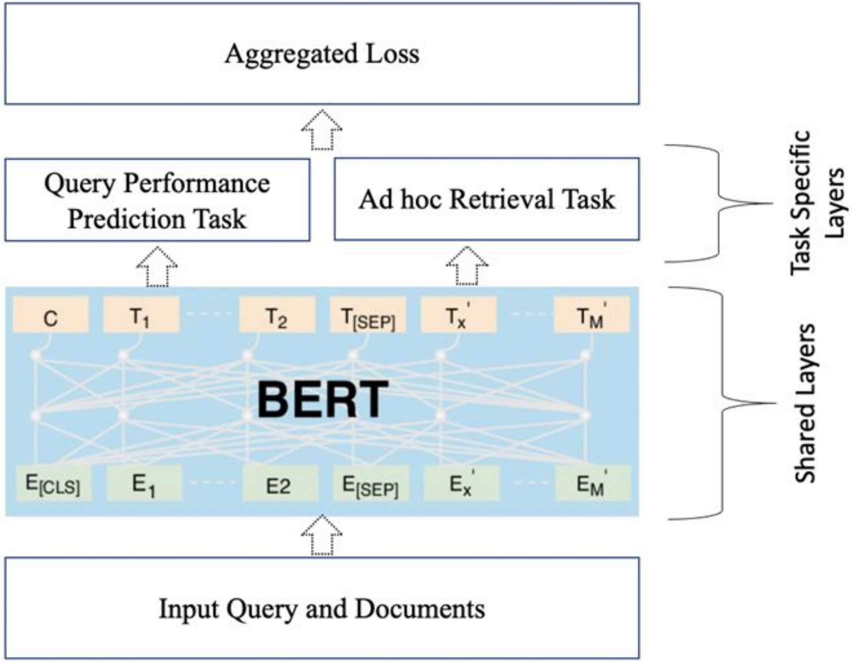
- ➢ Learning to rank and learning to predict query performance simultaneously will result in more effective ranking and more accurate performance prediction given the synergies between the two tasks.

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.



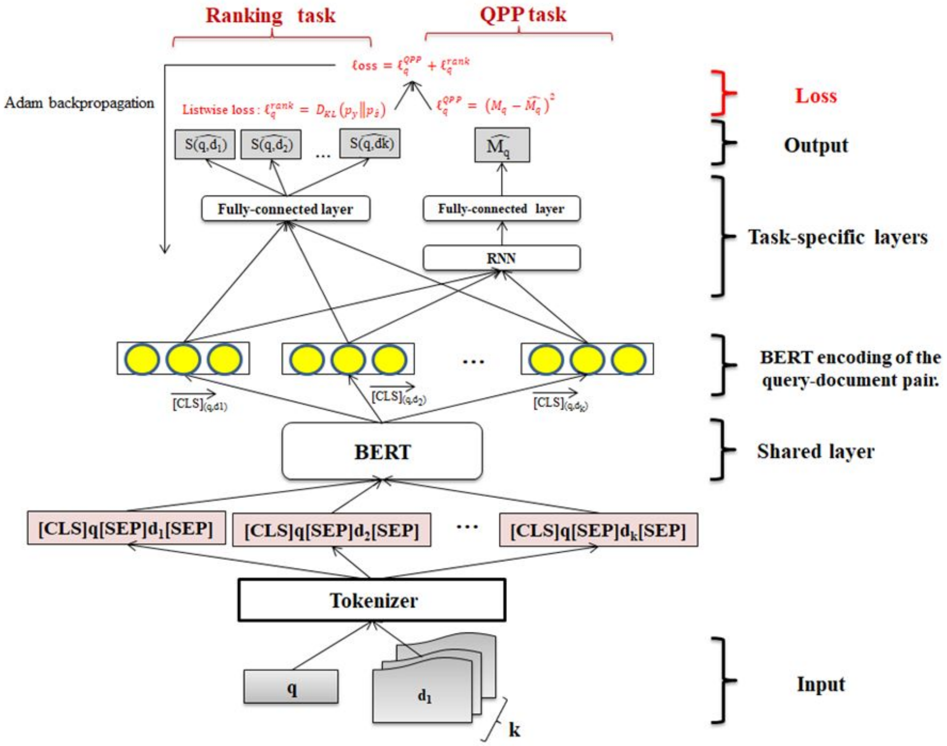
Learning to Rank and Predict

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 <u>listwise learning to rank paradigm.</u>

- ➢ M-QPPF simultaneously optimizes two different loss functions, one loss function for the document ranking task and another for the QPP task.

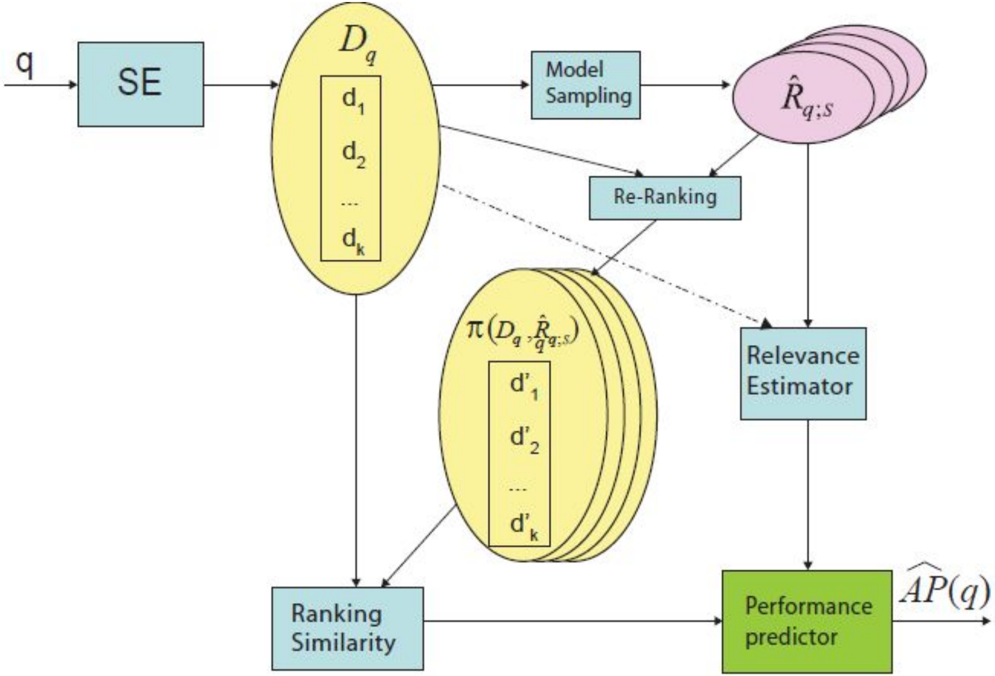


Utility Estimation Framework

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

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

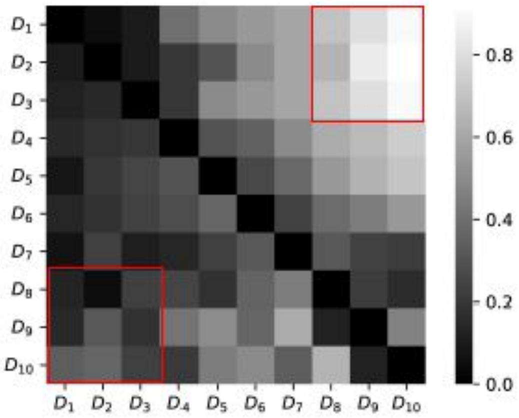
Approach: Predicting utility as the similarity between retrieved ranked list and an ideal ranked list.



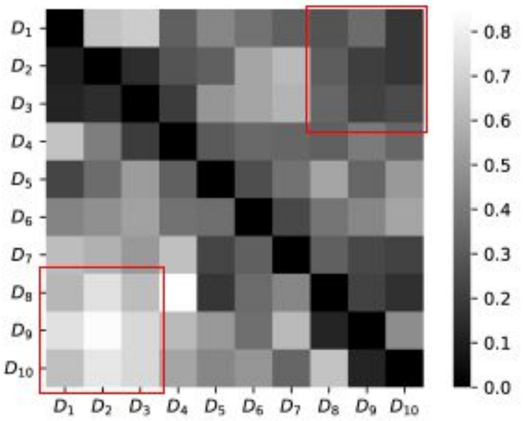
Utility Estimation Framework

Idea:Comparing the original list with a pairwise ranker

Approach: $\langle Q, D', D \rangle$ is used as an input, the pairwise model outputs a likelihood of D' being ranked better than D.



☺



☹

Q & A

QPP for various search scenarios

QPP for various search scenarios

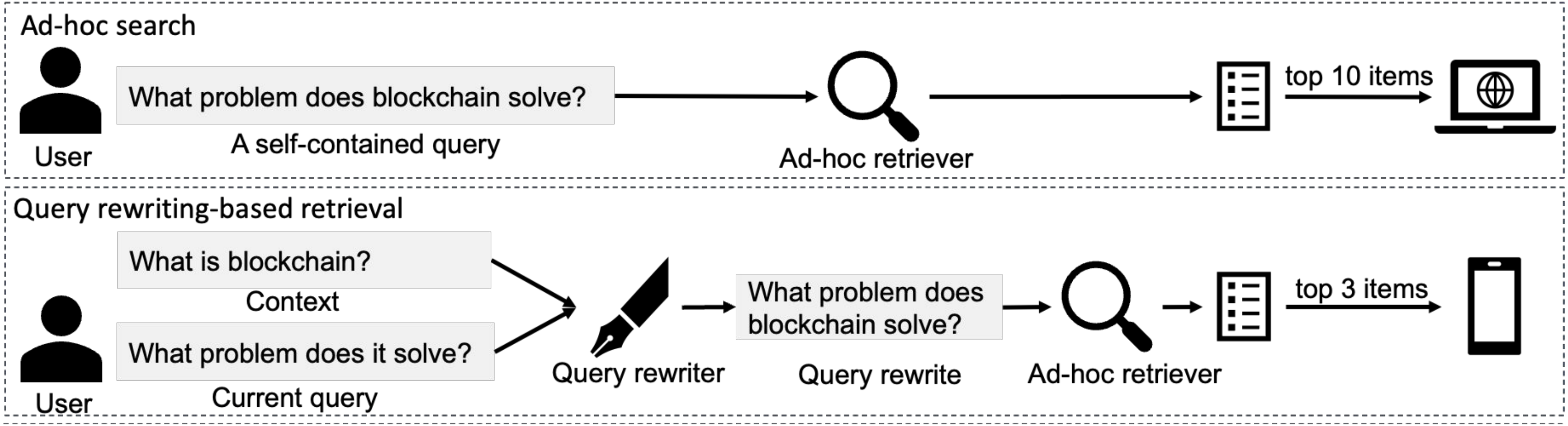
- QPP has been investigated in various scenarios:
    - Text search
        - Ad-hoc search
        - Conversational search
        - Open-domain question answering
    - Image search
        - Text-to-image search
        - Image-to-image search

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



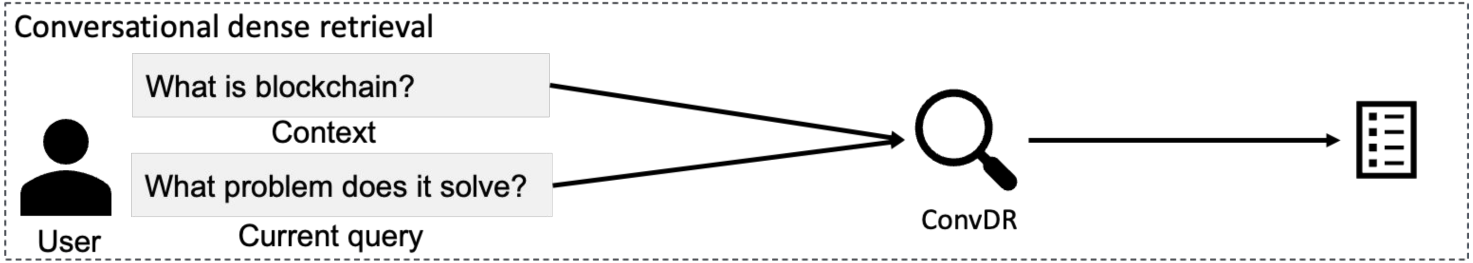
QPP for various search scenarios

- 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?

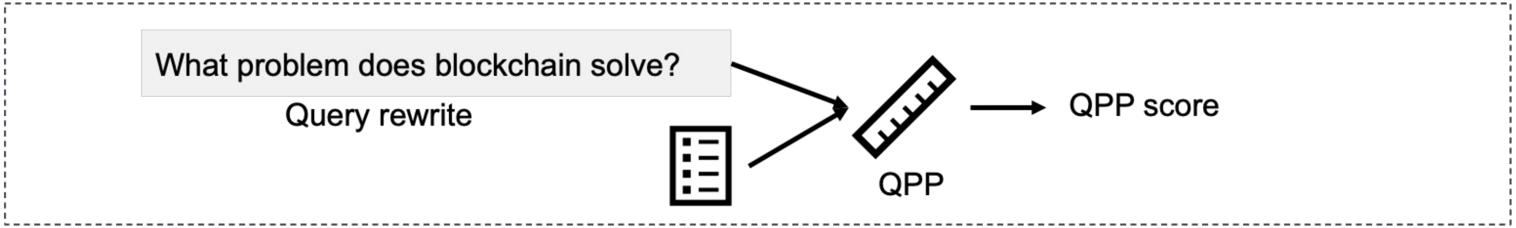
[1] Meng et al. Query Performance Prediction: From Ad-hoc to Conversational Search. SIGIR 2023..

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.

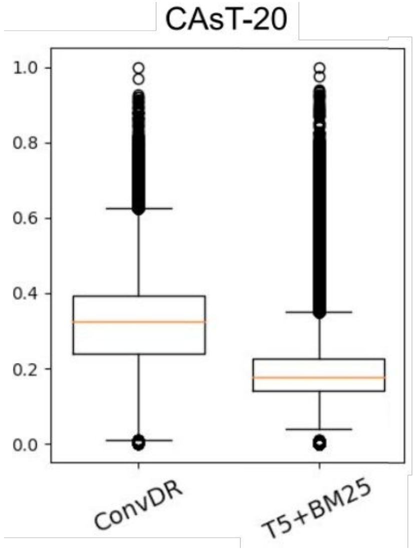
QPP for various search scenarios

- 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

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

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



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

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

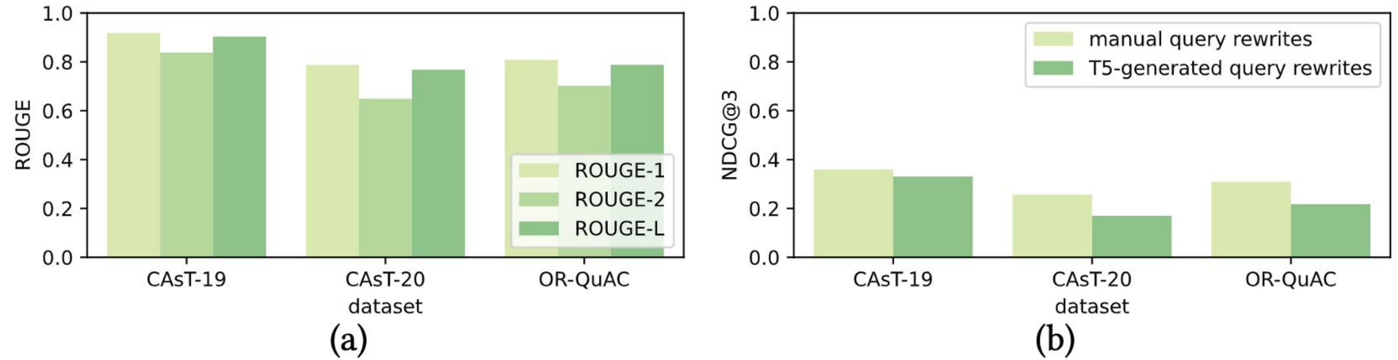


Figure 1: The similarity between manual and T5-generated query rewrites in terms of ROUGE (a) and the retrieval quality of BM25 for manual/T5-generated query rewrites in terms of NDCG@3 (b).

[1] Meng et al. Performance Prediction for Conversational Search Using Perplexities of Query Rewrites. QPP++ 2023..

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

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

    - [1] found that
      - PPL-QPP results in higher QPP quality, especially on datasets where query rewriting is challenging

[1] Meng et al. Performance Prediction for Conversational Search Using Perplexities of Query Rewrites. QPP++ 2023..

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

[1] Faggioli et al. A Geometric Framework for Query Performance Prediction in Conversational Search. SIGIR 2023.

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

[1] Hashemi et al. Performance Prediction for Non-Factoid Question Answering. ICTIR 2019.

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 anwer types
                - {Person, Organization, Location, Date, …}

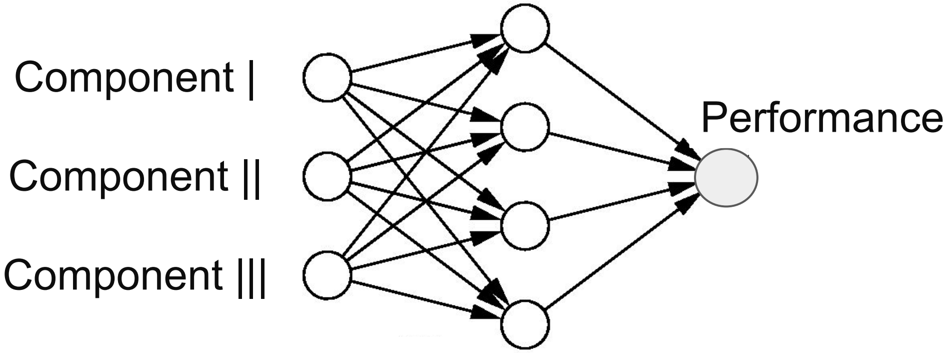
[1] Krikon et al. Predicting the Performance of Passage Retrieval for Question Answering. CIKM 2012.

[2] Hashemi et al. Performance Prediction for Non-Factoid Question Answering. ICTIR 2019.

Samadi et al. Performance Prediction for Multi-hop Questions. Axiv 2023.

QPP for various search scenarios

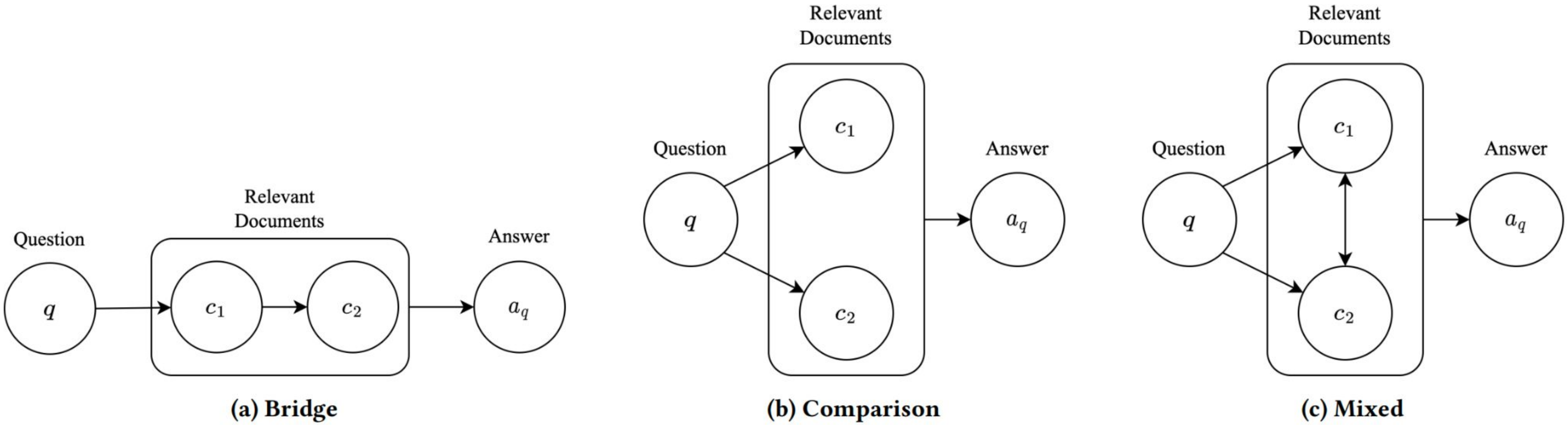
- QPP for open-domain question answering (QA)
  - [1] proposes a regression-based supervised QPP method
    - Aggregate three kinds of features:
      - Ranking scores
      - BERT(query)
      - BERT(query || answer 1), …, BERT(query || answer k)
    - Feed them into a fully-connected network producing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



[1] Hashemi et al. Performance Prediction for Non-Factoid Question Answering. ICTIR 2019.

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



(a) Bridge

(b) Comparison

(c) Mixed

[1] Samadi et al. Performance Prediction for Multi-hop Questions. Axiv 2023.

QPP for various search scenarios

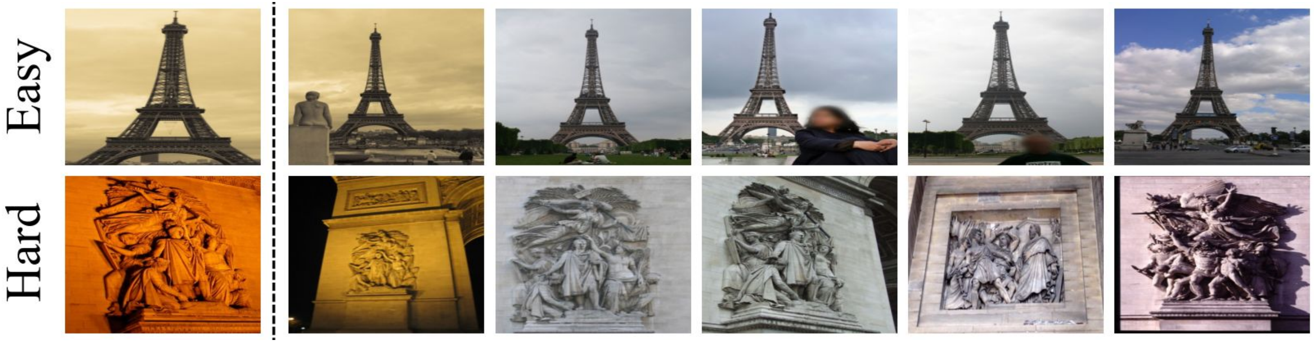
- QPP for image search
  - QPP for text-to-image search
    - [1] proposes adapted clarity scores, which measures the difference in the distribution of the retrieved images and the whole collection
    - [1] proposes adapted coherence scores, which measures the visual similarity among the retrieved images
    - [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



[1] Poesina et al. iQPP: A Benchmark for Image Query Performance Prediction. SIGIR 2023.

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
        - IR system configuration selection
        - Ranker selection
        - Fusion-based retrieval
        - Candidate generation
    - Others
        - Action prediction
        - Conversation contextualization
        - Query routing
        - Query-specific pool depth prediction

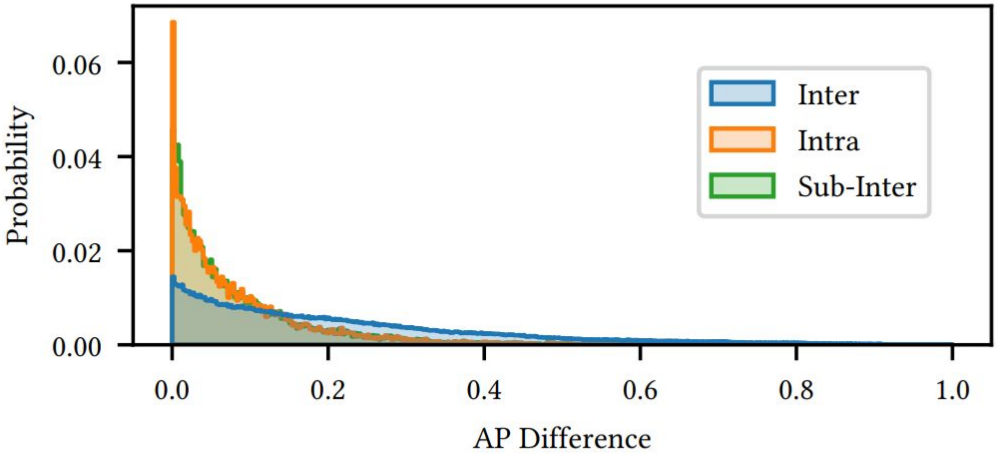
Applications of QPP

- 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

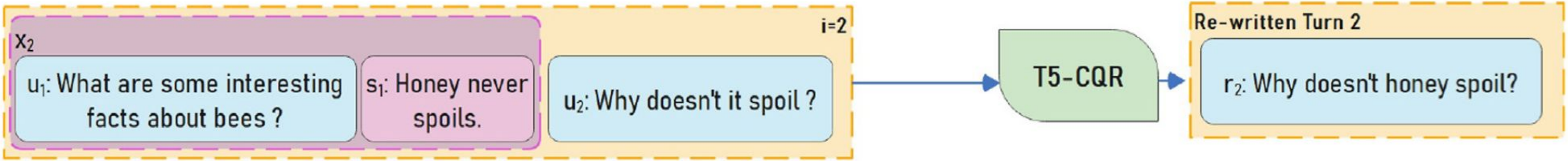
- Query variant selection
  - [1] reveals the reason:
    - Actual effectiveness differences among query variants are smaller than those among topics.



[1] Zendel et al. Is Query Performance Prediction With Multiple Query Variations Harder Than Topic Performance Prediction? SIGIR 2021.

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

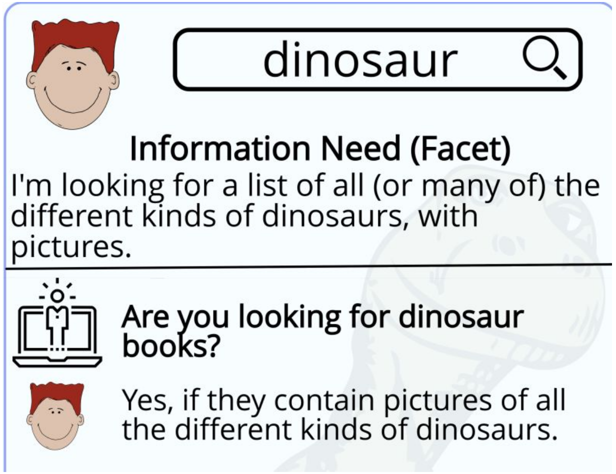


X_2
i=2
u_1: What are some interesting facts about bees ?
s_1: Honey never spoils.
u_2: Why doesn't it spoil ?
T5-CQR
Re-written Turn 2
r_2: Why doesn't honey spoil?

[1] Al-Thani et al. Improving Conversational Search with Query Reformulation Using Selective Contextual History. Data and Information Management 2023.

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]



Information Need (Facet)

I'm looking for a list of all (or many of) the different kinds of dinosaurs, with pictures.

Are you looking for dinosaur books?

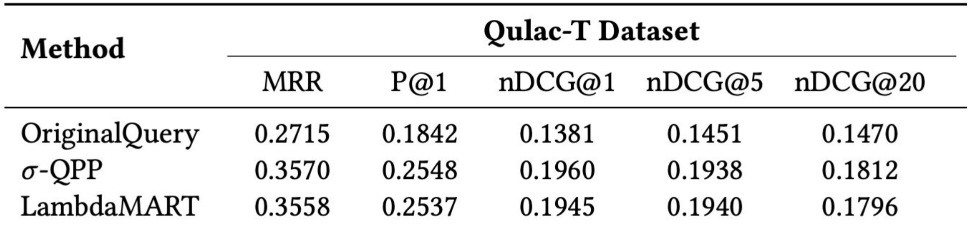
Yes, if they contain pictures of all the different kinds of dinosaurs.

[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
σ-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.

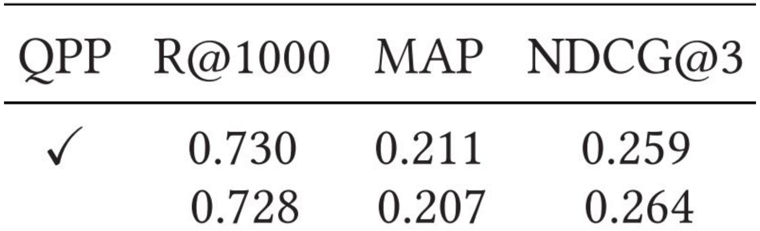
Applications of QPP

- 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

[1] Lin et al. Multi-stage Conversational Passage Retrieval: An Approach to Fusing Term Importance Estimation and Neural Query Rewriting. TOIS 2021.

Applications of QPP

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

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.

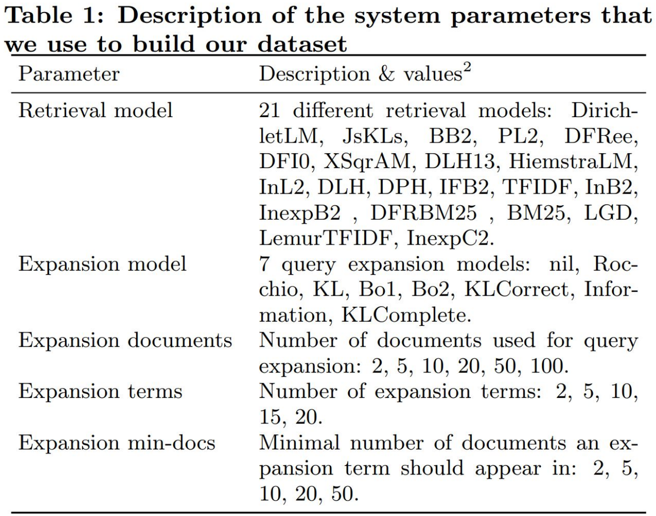


Table 1: Description of the system parameters that we use to build our dataset

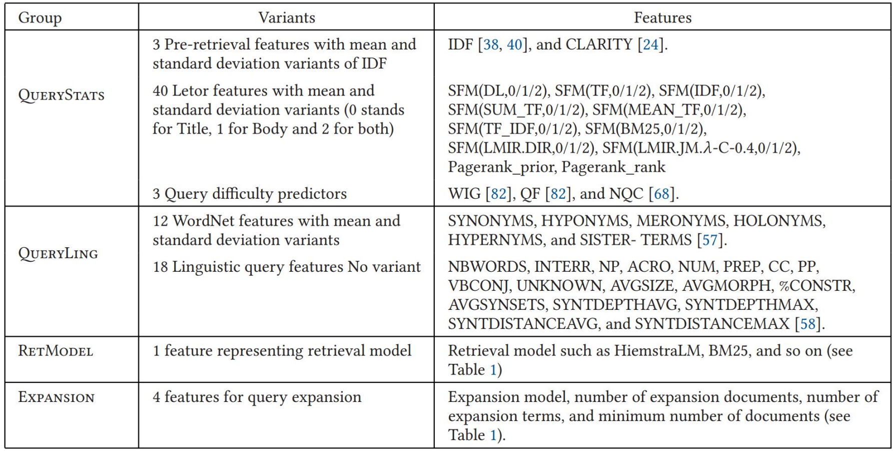
Parameter	Description & values[2]
Retrieval model	21 different retrieval models: DirichletLM, JsKLs, BB2, PL2, DFRee, DFI0, XSqrAM, DLH13, HiemstraLM, InL2, DLH, DPH, IFB2, TFIDF, InB2, InexpB2 , DFRBM25 , BM25, LGD, LemurTFIDF, InexpC2.
Expansion model	7 query expansion models: nil, Rocchio, KL, Bo1, 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	IDF [38, 40], and CLARITY [24].
	40 Letor features with mean and standard deviation variants (0 stands for Title, 1 for Body and 2 for both)	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.λ-C-0.4,0/1/2), Pagerank_prior, Pagerank_rank
	3 Query difficulty predictors	WIG [82], QF [82], and NQC [68].
QueryLing	12 WordNet features with mean and standard deviation variants	SYNONYMS, HYPONYMS, MERONYMS, HOLONYMS, HYPERNYMS, and SISTER- TERMS [57].
	18 Linguistic query features No variant	NBWORDS, INTERR, NP, ACRO, NUM, PREP, CC, PP, VBCONJ, UNKNOWN, AVGSIZE, AVGMORPH, %CONSTR, AVGSYNSETS, SYNTDEPTHAVG, SYNTDEPTHMAX, SYNTDISTANCEAVG, and SYNTDISTANCEMAX [58].
RetModel	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

[1] Khramtsova et al. Leveraging LLMs for Unsupervised Dense Retriever Ranking. arXiv 2023.

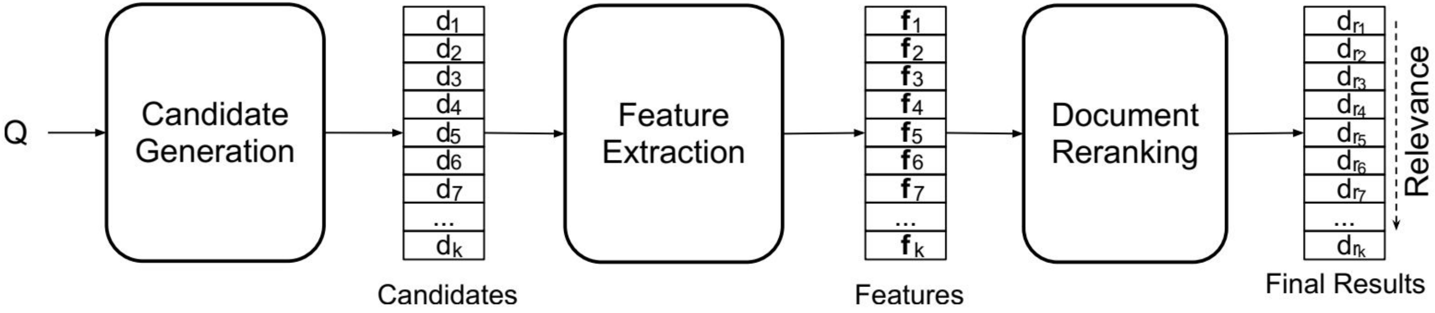
Applications of QPP

- 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

[1] Raiber et al. Query-Performance Prediction: Setting the Expectations Straight. SIGIR 2014.

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



[1] Tonellotto et al. Efficient and Effective Retrieval using Selective Pruning. WSDM 2013.

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

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

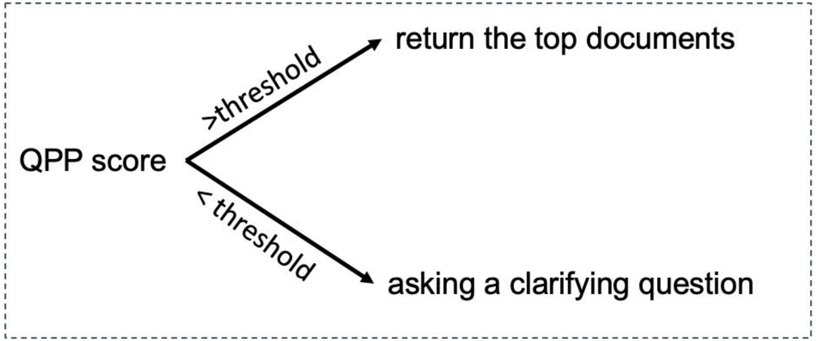
[1] Tonellotto et al. Efficient and Effective Retrieval using Selective Pruning. WSDM 2013.

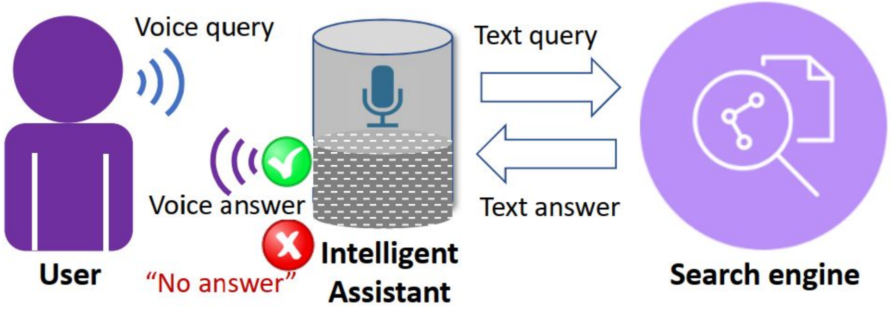
Applications of QPP

- QPP has been applied to various downstream scenarios:
  - ~~Query-oriented~~
    - ~~Query variant selection~~
    - ~~Clarifying question selection~~
    - ~~Selective query expansion~~
  - ~~Ranker-oriented~~
    - ~~IR system configuration selection~~
    - ~~Ranker selection~~
    - ~~Fusion-based retrieval~~
    - ~~Candidate generation~~
  - Other
    - 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-tined BERT
  - [2] use a set of QPP features to train a classier
    - QPP features make a difference



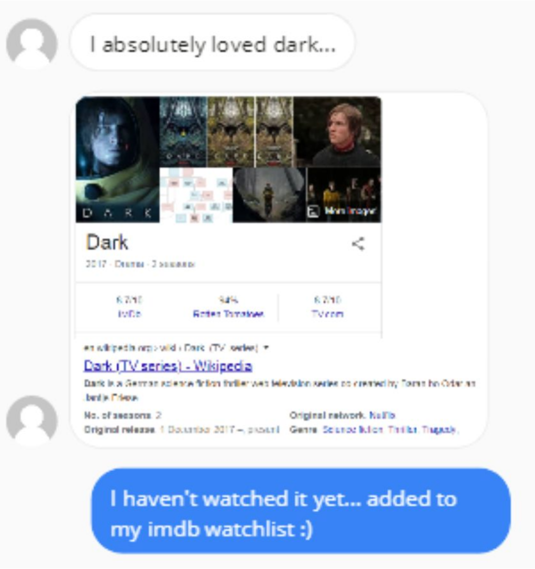


[1] Arabzadeh et al. Unsupervised Question Clarity Prediction Through Retrieved Item Coherency. CIKM 2022.

[2] Roitman et al. A Study of Query Performance Prediction for Answer Quality Determination. ICTIR 2019.

Applications of QPP

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



I absolutely loved dark...

Dark

2017 · Drama · 2 seasons

8.7/10	94%	8.7/10
IMDb	Rotten Tomatoes	TV.com

en.wikipedia.org › wiki › Dark_(TV_series) ▾

Dark (TV series) - Wikipedia

Dark is a German science fiction thriller web television series co-created by Baran bo Odar and Jantje Friese.

No. of seasons: 2

Original release: 1 December 2017 – present

Original network: Netflix

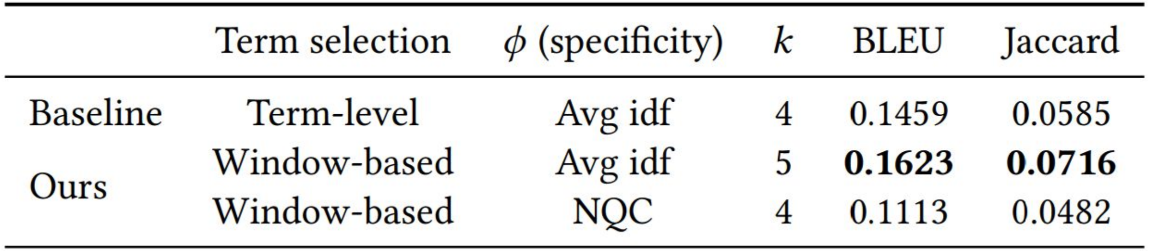
Genre: Science fiction, Thriller, Tragedy,

I haven't watched it yet... added to my imdb watchlist :)

[1] Pal et al. Effective Query Formulation in Conversation Contextualization: A Query Specificity-based Approach. ICTIR 2021.

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

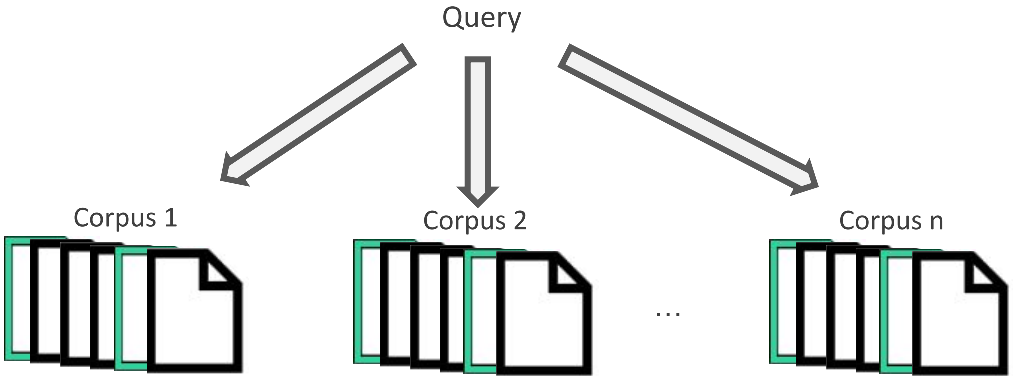


	Term selection	ϕ (specificity)	k	BLEU	Jaccard
Baseline	Term-level	Avg idf	4	0.1459	0.0585
Ours	Window-based	Avg idf	5	**0.1623**	**0.0716**
	Window-based	NQC	4	0.1113	0.0482

[1] Pal et al. Effective Query Formulation in Conversation Contextualization: A Query Specificity-based Approach. ICTIR 2021.

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



[1] Khramtsova et al. Query-performance Prediction for Effective Query Routing in Domain-specific Repositories. JASIST 2014.

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

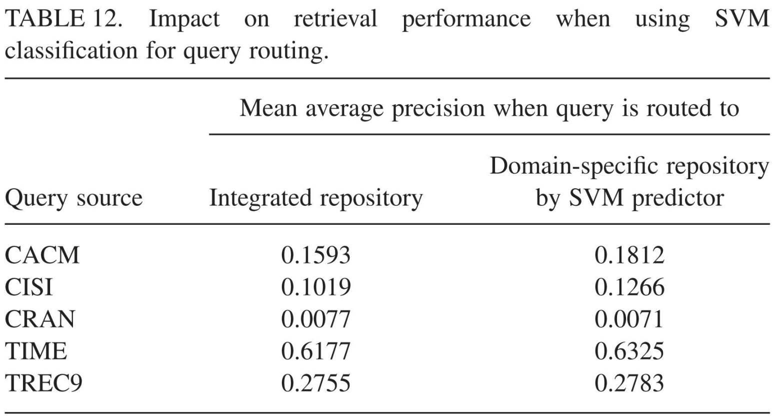


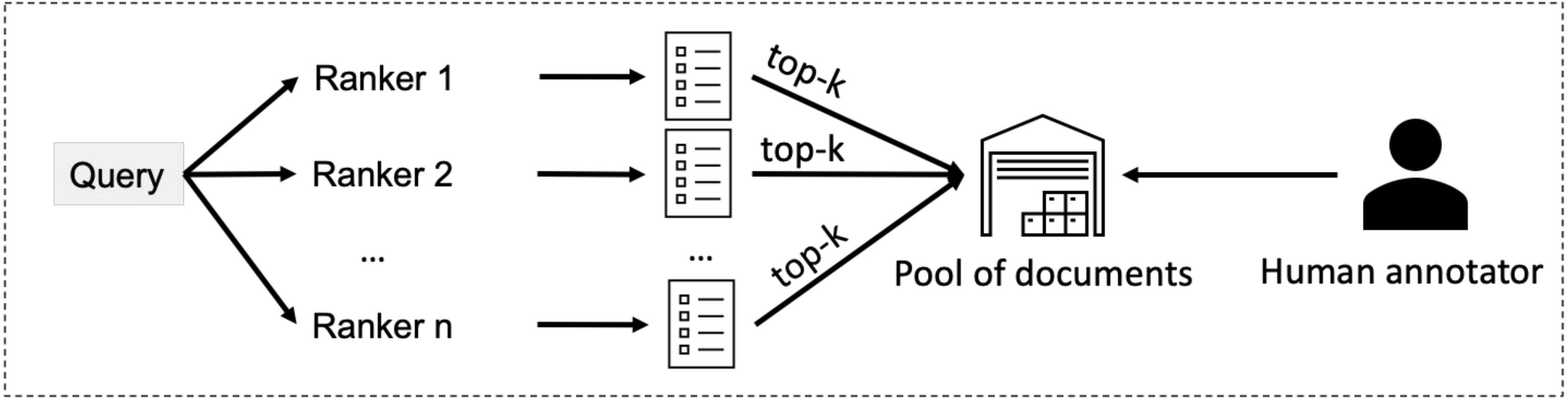
TABLE 12. Impact on retrieval performance when using SVM classification for query routing.

	Mean average precision when query is routed to	
Query source	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

[1] Khramtsova et al. Query-performance Prediction for Effective Query Routing in Domain-specific Repositories. JASIST 2014.

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



[1] Ganguly et al. Query-specific Variable Depth Pooling via Query Performance Prediction. SIGIR 2023.

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

[1] Ganguly et al. Query-specific Variable Depth Pooling via Query Performance Prediction. SIGIR 2023.

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

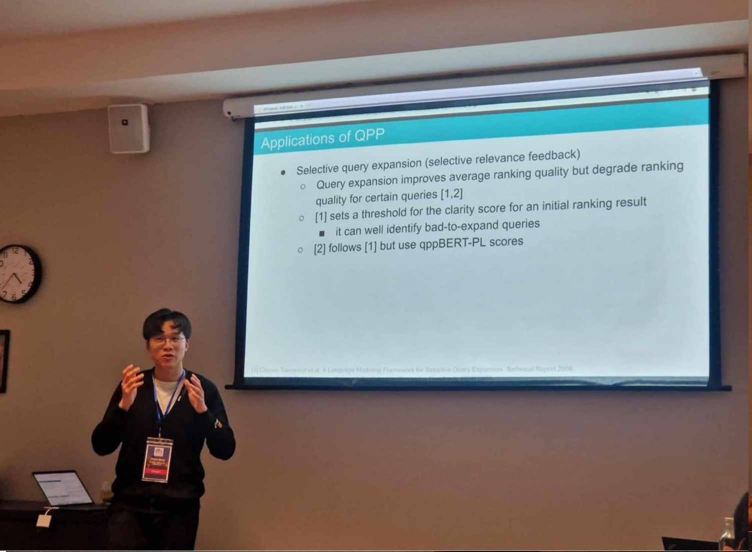
- Limitation and future directions:
  - Existing QPP approaches typically predict only a single real-valued score, and do not require the predicted score to approximate a specific IR metric
    - Relying on a single value to represent different IR metric leads to a ``one size fits all'' issue; Some metrics do not correlate well. Existing regression-based supervised QPP methods need to train separate models for different metrics
    - A single-score prediction limits the interpretability of QPP

Conclusions and future directions

- Limitation and future directions:
    - QPP has limited performance on some downstream tasks because the target mismatch between QPP objective and downstream task evaluation

Conclusions and future directions

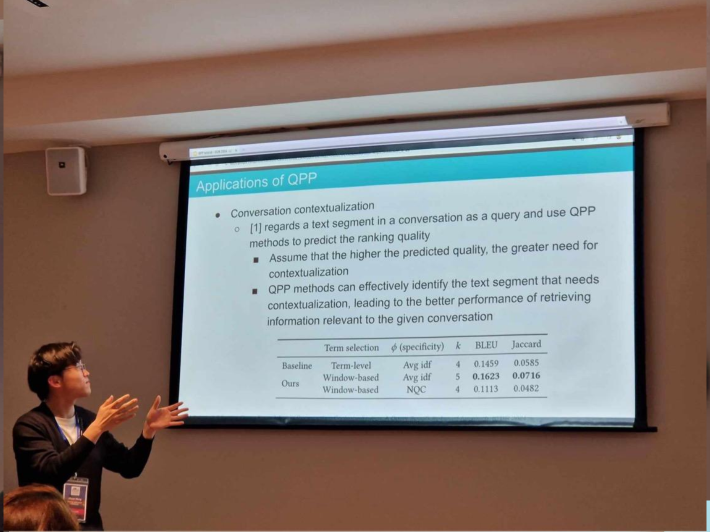
- Limitation and future directions:
    - QPP can only be used for predicting the performance of ranking-based systems. How to predict the performance of generative systems



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.



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	ϕ (specificity)	k	BLEU	Jaccard
Baseline	Term-level	Avg idf	4	0.1459	0.0585
Ours	Window-based	Avg idf	5	**0.1623**	**0.0716**
	Window-based	NQC	4	0.1113	0.0482

Thank you

Discussions