Discovering Biases in Information Retrieval Models Using Relevance Thesaurus as Global Explanation

Youngwoo Kim, Razieh Rahimi, and **James Allan**
University of Massachusetts Amherst
{youngwookim, rahimi, allan}@cs.umass.edu

Abstract

Most efforts in interpreting neural relevance models have focused on local explanations, which explain the relevance of a document to a query but are not useful in predicting the model's behavior on unseen query-document pairs. We propose a novel method to globally explain neural relevance models by constructing a "relevance thesaurus" containing semantically relevant query and document term pairs. This thesaurus is used to augment lexical matching models such as BM25 to approximate the neural model's predictions. Our method involves training a neural relevance model to score the relevance of partial query and document segments, which is then used to identify relevant terms across the vocabulary space. We evaluate the obtained thesaurus explanation based on ranking effectiveness and fidelity to the target neural ranking model. Notably, our thesaurus reveals the existence of brand name bias in ranking models, demonstrating one advantage of our explanation method. [1]

1 Introduction

Transformer-based information retrieval (IR) models (Dai and Callan, 2019; MacAvaney et al., 2019) that are trained on large datasets like MS MARCO (Nguyen et al., 2016) are very effective in predicting relevance between a query and document. Contextual representations in these models enable semantic matching, such as matching the query term "car" with the document term "vehicle". However, it is challenging for researchers to predict the potential failures of a model, such as when it matches a query term to non-relevant document terms.

Query term	Document term			
injury	injure 0.26	wound 0.24	torn 0.19	...
when	24th 0.33	2010 0.11	2015 0.01	...
car	vehicle 0.68	ford 0.38	honda 0.28	...
cud	cudâ 0.50	cuda 0.50	...	

Table 1: Example entries from our relevance thesaurus. The numbers indicate the degree of relevance. Unexpected behaviors found by our method are highlighted.

Another potential risk associated with neural retrieval models is an unintended bias toward certain entities or groups (May et al., 2019). While it is appropriate for a model to associate the query term "car" with various car brand names (e.g., Ford), the model should not exhibit a strong preference for a particular brand, leading to the model favoring that brand over another when all other factors are identical. For the safe deployment of information retrieval (IR) models in real-world scenarios, detailed global understanding of model behavior are essential, such as providing which lexical expressions are considered relevant by the models.

To address these challenges and mitigate potential risks, post-hoc explanation methods for black-box machine learning models can be employed. Most explanations for IR model explanations are local explanations, focusing on individual model predictions, such as a specific query-document pair (Kim et al., 2022) or a ranked list for a query (Verma and Ganguly, 2019; Llordes et al., 2023). These explanations indicate which terms in the documents contribute to its relevance to the query. However, local explanations have two major limitations that hinder their ability to infer cases where the model may exhibit unexpected behavior.

First, the explanations are limited to the terms observed in the given query and document, and bi-

[1] Code and results are available at https://github.com/youngwoo-umass/RelevanceThesaurus

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 19530–19547
November 12-16, 2024 ©2024 Association for Computational Linguistics

ases may exist in queries or documents that were not evaluated or inspected with the explanations. Second, attribution to document terms by explainers may be highly dependent on the contexts of those terms, therefore it is unclear whether the attributed document terms in other contexts would match the query.

To overcome these limitations, we propose building a global explanation (Guidotti et al., 2018) that provides lexical insights about query-document terms that are matched by the model in all contexts. We can describe a model's behavior in a compact and interpretable structure that is not limited to a specific instance.

Our proposed global explanation focuses on identifying relevant pairs of query and document terms that can effectively explain the matching behavior of neural retrieval models. We refer to this format of explanation as a *relevance thesaurus*, with examples illustrated in Table 1. The table indicates that, if a query contains term "injury," then it is likely for the model to match the query term with document terms "injure," or "wound," with the former being the more likely. This allows researchers to anticipate which terms, when present in a document, would lead the model to predict higher relevance for that document, without requiring additional context from the document.

Constructing a relevance thesaurus is challenging due to the large number of potential term pairs. Many local (Ribeiro et al., 2016) and global explanation (Han et al., 2020) methods build a candidate set of features from data and adjust their scores based on the target model's outputs. However, this approach becomes infeasible when the number of features reaches to billions, as in our study. To overcome this challenge, we propose a novel approach that distill the knowledge of the target model into an intermediate neural model, PaRM (Partial Relevance Model), which is then used to infer important features.

PaRM is designed to predict a score for a term pair, which is then used to predict the score for the corresponding query-document pair. By training PaRM with knowledge distillation from the target neural model to be explained, we ensure that the generated relevance thesaurus faithfully explains the target model's behavior.

Rather than assessing the accuracy of each term pair in the relevance thesaurus individually, the thesaurus is extrinsically evaluated by integrating it into lexical matching models (BM25 (Robertson et al., 2009) and QL (Ponte and Croft, 1998)), adding interpretable semantic matching to them. The resulting retrieval methods are evaluated based on retrieval effectiveness and fidelity to the target neural retrieval models. The results on multiple datasets show the effectiveness of the acquired relevance thesaurus.

To demonstrate the advantages of our relevance thesaurus, we introduce three unexpected findings about the behavior of neural retrieval models trained on MS MARCO, obtained from our analysis of the relevance thesaurus: (1) the *car-brand bias*, which suggests that models exhibit biases towards certain car brands; (2) the *temporal bias*, which indicates that models consider distant future or past years to be more strongly associated with the query term "when" compared to the current year; (3) the *postfix-a* finding, which reveals that models treat the character "a" appended to a term as equivalent to a quotation mark due to encoding errors.

Experiments using multiple state-of-the-art neural information retrieval models demonstrate that these behaviors are not limited to the cross-encoder ranker which is used to distill the relevance thesaurus but are also the case in multiple other IR models, Splade (Formal et al., 2021b) and Contriever (Izacard et al., 2021). This highlights the importance of global explanations for retrieval models.

2 Related works

2.1 Global model explanations

Large portions of works on global explanations are for classification tasks on tabular features (Craven and Shavlik, 1995; Boz, 2002; Guidotti et al., 2018). They cannot be applied to the Transformer model for token sequences, as tabular features are not defined. Instead, global explanation works in the NLP domain target single text classification, by attributing output labels to some words or phrases (Rajagopal et al., 2021; Han et al., 2020). However, these word-to-output label attributions are not applicable for explaining text pair models like IR models, because document terms' importance is highly dependent on queries. It would make a more meaningful explanation if it indicate certain terms or phrases from the query are associated with specific terms or phrases that appear in the document.

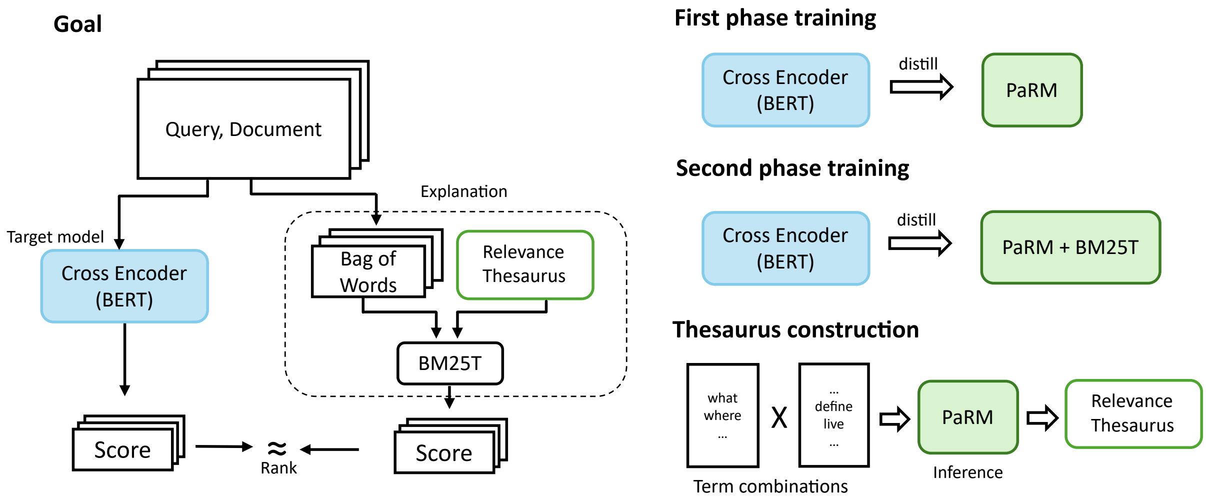


Figure 1: Our goal is to build a relevance thesaurus that can approximate the cross-encoder model (left). The relevance thesaurus is expected to be generalizable to any queries. The figure on the right shows how the relevance thesaurus is constructed. The colored boxes are black-box models, and the white boxes are interpretable components.

2.2 Explanations for neural IR

Existing neural IR models (Khattab and Zaharia, 2020; Gao et al., 2021; Formal et al., 2021b; Nogueira et al., 2019; Kim et al., 2021) encode entire queries and/or documents with a single Transformer network and are not capable of encoding parts of the query/document in the absence of the remaining context. Applying perturbation-based explanation approaches (Kim et al., 2020; Ribeiro et al., 2016) on these models can be problematic because removing tokens from a query could lead to a larger change of the meaning in IR tasks than the other NLP tasks. For example, a document that is relevant to the query "ACL location" is not relevant to the query "location", as the relevant document for this query needs to describe what the expression "locations" means rather than mentioning any location.

Existing IR models explanations (Verma and Ganguly, 2019; Llordes et al., 2023; Lyu and Anand, 2023; Pandian et al., 2024; Naseri et al., 2021) mostly work in the query-level, and output terms for one query cannot be used to infer the model's behaviors in other queries. Chowdhury et al. (2023) target explaining categorical features in learning-to-rank IR models, while we target sequence processing Transformer models.

To enhance lexical models with recent advances, Boytsov and Kolter (2021) proposed fine-tuning BERT (Devlin et al., 2019) for the translation language model (Berger and Lafferty, 1999). This approach, however, is limited to the semantic matches between terms in BERT's subword vocabulary and does not extend to terms formed from multiple subwords. Moreover, the work lacks analysis or evaluation regarding the explanation perspectives and does not provide qualitative insights from the outcomes.

2.3 Interpretable NLP models

Our proposed model architecture is motivated by the series of the work on the natural language inference task (Wu et al., 2021; Stacey et al., 2022; Kim et al., 2023). Specifically, we adopted the idea of partitioning a sentence into two segments from the work by Kim et al. (2023).

3 Relevance thesaurus building

We define the global explanation of an information retrieval model, based on the definition by Guidotti et al. (2018), as follows:

Model explanation problem. Given a black-box relevance predictor S_b that takes a query q and a document d as inputs and predicts a relevance score $y \in Y$, a global model explainer aims to find a human-interpretable explanation E and an explanation logic ϵ. The explanation logic ϵ is a function that converts the explanation E into a global predictor S_e. The global predictor S_e predicts a score $\hat{y}$ for (q, d), which approximates the prediction y.

As a black-box predictor S_b to be explained, we target the full cross-encoder (CE) document ranking model (Dai and Callan, 2019), which takes the concatenated sequence $q; d$ into the Transformer encoder to predict the relevance. As a format for an explanation E, we use a relevance thesaurus, which is a set of triplets (qt, dt, s), where qt is a query term, dt is a document term, and s is the score assigned to the term pair.

To build an interpretable predictor S_e, we incorporate the relevance thesaurus (E) into the BM25 scoring function to address vocabulary mismatches between queries and documents. The relevance thesaurus captures semantic relationships between terms that may not be explicitly present in the query or document. Incorporating the relevance thesaurus improves the retrieval performance of the sparse retrieval model BM25, and provides an interpretable explanation of the CE model's behavior.

Many local and global explanation methods explicitly build a candidate set of features (e.g., terms) from data. To determine the appropriateness of each candidate feature as an explanation, these methods initially assign scores to the features. Then, the scores are adjusted based on the observed behaviors of the model. This strategy maintains explicit feature candidates and their scores during the optimization process. This can be challenging as the number of features increases, especially in our explanation format where the number of term pairs can scale to billions.

To address this challenge, we propose implicitly optimizing the explanation features using an intermediate neural model that scores features, namely term pairs.

As an intermediate neural model, we propose PaRM (Partial Relevance Model), which is designed to score relevance between partial segments of a query and document. Unlike the original cross-encoder model and other relevance models that assign a score to an entire query and document, PaRM can predict meaningful scores for partial queries or documents.

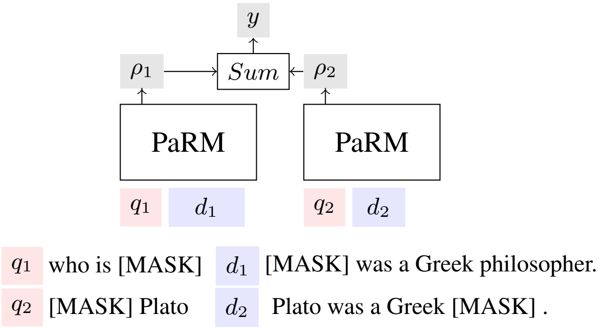


Figure 2: The first phase of training PaRM. The query "who is Plato" (red) is partitioned into q_1 and q_2. The document "Plato was a Greek philosopher" (blue) is masked to generate d_1 and d_2.

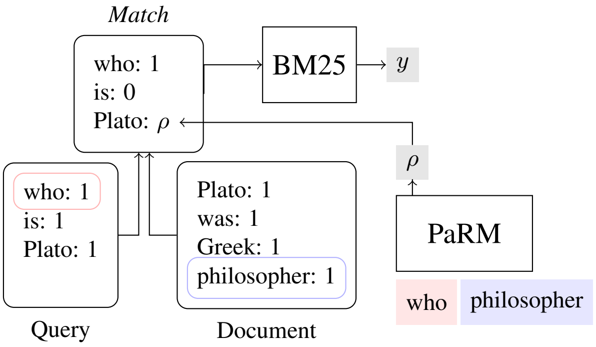


Figure 3: The second phase of training PaRM. BM25 computes a relevance score based on the frequency of each query term in the document (*Match*). If a query term (e.g., who) does not appear in the document, the PaRM score ρ for the most relevant document term is used as a discounted query term frequency.

This is important because the original CE model cannot accurately assess the contributions of individual tokens when they are isolated from their original contexts. For example, if the query is "Who is Plato" and the document is a single term "Plato", the original CE model would likely predict a score indicating non-relevance, as a document with a single term is unlikely to provide meaningful information. PaRM, on the other hand, predicts a score that indicates partial relevance, which can be combined with partial relevance scores for other terms to build the final relevance score for the query-document pair. We are using the context independence assumption here. This assumption is useful because it allows us to use any term pair predicted as relevant by PaRM to be globally indicative of relevance, which is not possible using local explanations.

PaRM is trained end-to-end by distilling predictions from the CE model. However, a challenge arises because PaRM is expected to predict a score for a query term and document term, while the available signal is only at the query-document level. To supervise term-pair level scores in PaRM using the CE model, alignments between query and document terms are required, but these are not directly available. The novelty of PaRM training lies in the ability to infer alignments in an unsupervised way.

In the first stage, we train PaRM to predict scores for two partial query-document segment pairs, using weak alignments, such as attention scores. After the first stage, we use the PaRM model to infer term-level alignments. In the second stage, we use the generated alignments by PaRM to further fine-tune the PaRM model to predict appropriately scaled scores for word-level relevance. These prediction are then used to create the final explanations (E).

3.1 PaRM first phase training

The first phase PaRM predicts the relevance score S_e for a given query-document pair by generating scores for two partial inputs, (q_1, d_1) and (q_2, d_2). These inputs are built by extracting a continuous span from the query q to form q_1 and using the remaining tokens with a [MASK] token for q_2.

The corresponding document segments, d_1 and d_2, are constructed by masking tokens from the document d that have low attention scores for q_1 and q_2, as determined by the full cross-encoder (CE) ranker.

We randomly select how many tokens to be left in d_i, ranging from one to all tokens of d (see Appendix D for details). In most cases, d_i contains sufficient evidence to learn relevance while still allowing for a few extreme cases where either only a single query term or a single document term is present.

Scores for (q_1, d_1) and (q_2, d_2) are obtained by projecting BERT's CLS pooling representations.

$$\text{PaRM}(q_i, d_i) = W \cdot BERT_{CLS}(q_i; d_i) + b \quad (1)$$

The final score for the query and document pair is the sum of the scores from two partial views.

$$S_e(q, d) = \text{PaRM}(q_1, d_1) + \text{PaRM}(q_2, d_2) \quad (2)$$

The combined score S_e is trained from the scores (S_b) of the target black-box model (CE) using margin mean square error (MSE) loss (Hofstätter et al., 2021) on relevant and non-relevant query-document pairs.

$$\begin{aligned}\mathcal{L} = \text{MSE}(&S_e(q, d^+) - S_e(q, d^-), \\ &S_b(q, d^+) - S_b(q, d^-))\end{aligned} \quad (3)$$

Once PaRM is trained, we can use it to score an arbitrary query span or document span, including a single term. However, the scores are only trained for ranking and not calibrated to a specific range, which makes it hard to determine which term pairs have a sufficiently large score to be included in the relevance thesaurus.

3.2 Fine-tuning PaRM with BM25

In the second phase, we fine-tune PaRM so that it scores the relevance of a query term qt and a document term dt on a scale from 0 to 1. Specifically, we consider the scenario of augmenting BM25 by handling vocabulary mismatch based on the scores from PaRM.

We consider a query-document pair that any query term is missing in the document. We assume that the document term that has the highest PaRM score against the corresponding query term is most likely to be relevant to the query term if any term is relevant. We then use the output of PaRM to replace the term frequency (Figure 3). If the assumed pair is relevant, it will be more likely to appear in the relevant document and will be trained to score higher, and non-relevant ones will appear in the non-relevant document and be trained lower.

For a pair of query q and document d, if any query term does not have an exact match in the document, we randomly select one query term qt to be trained. All document terms are scored against qt using PaRM(qt, dt) and the document term dt with the highest score is paired with qt. Note that terms are not from the BERT tokenizer, but are from the tokenizer developed for BM25. Thus, a single term can contain multiple BERT subwords.

The training network is defined as follows. To ensure the output is between 0 to 1, we apply a sigmoid layer (σ) on top of the projected output.

$$\text{PaRM}(qt, dt) = \sigma(W \cdot \text{BERT}_{\text{CLS}}(qt; dt) + b) \quad (4)$$

In the original BM25 formula, the score for the query term qt is determined by qt's document frequency, $tf_{qt,d}$. We modify BM25 so that when a query term does not appear in the document, $tf_{qt,d}$ is replaced with the output of PaRM(qt, dt).

$$f(qt, d) = \begin{cases} tf_{qt,d} & \text{if } qt \in d \\ \text{PaRM}(qt, dt) & \text{if } qt \notin d \end{cases} \quad (5)$$

Note that $tf_{qt,d}$ can be large but PaRM is bounded above by 1, thus a non-exact match is never stronger than a single exact match. The relevance score is computed based on the BM25 scoring function:

$$S_e(q,d) = \sum_{qt \in d} \text{IDF}(qt) \cdot \frac{f(qt,d)\cdot(k_1+1)}{f(qt,d)+K} \tag{6}$$

where K is a function of document lengths, which is independent of $f(qt,d)$.

PaRM is trained end-to-end from the pairwise hinge loss between a relevant pair (q,d^+) and a non-relevant pair (q,d^-):

$$\mathcal{L} = \max\left(0, 1 - S_e(q,d^+) + S_e(q,d^-)\right). \tag{7}$$

Note that we do not use knowledge distillation here, because the output scores scale of the BM25 scoring function is not easily adjustable and may not be possible to match the score margin of the neural ranking model.

During the training phase, equations 4 to 7 are implemented within a neural network framework, and the gradient to the loss $\mathcal{L}$ is back-propagated to train PaRM's parameters. Note that PaRM scores for selecting the highest scored dt are pre-computed with the model after the first phase.

After PaRM is fine-tuned, it can pre-compute the scores for potential qt and dt candidates. These candidate pairs and scores compose a relevance thesaurus. The acquired relevance thesaurus can be used to either inspect the model's behavior or used with the BM25 scoring function.

We name the modified BM25 function that addresses non-exact lexical matches based on the relevance thesaurus as BM25T (BM25 with Thesaurus). For each query term qt, if qt is found in the document d, relevance thesaurus is not used. If qt is not found, the document term $dt \in d$ with the highest (pre-computed) PaRM(qt, dt) score in the thesaurus is used to compute the score.

4 Experiments

4.1 Implementation

As a target ranker to be explained, we use a publicly available cross-encoder, which is fine-tuned from distilled-BERT[2]. The predictions of this model are used as teacher scores in Equation 3. We initialized PaRM with pre-trained BERT-based-uncased. The maximum sequence length of the input in the first and second phases of training PaRM is set to 256 and 16 tokens, respectively.

The models are trained on the widely used MS MARCO passage ranking dataset (Nguyen et al., 2016). BM25 and BM25T use the tokenizer from the Lucene library [3] with the Krovetz stemmer (Krovetz, 1993), preferred over the Porter stemmer (Porter, 1980) for producing actual words.

Relevance thesaurus construction PaRM scores the candidate term pairs to build the final relevance thesaurus as a global explanation of the full cross-attention ranking model. The candidates are drawn from the frequent terms in the MS MARCO corpus. The top 10K frequent terms were considered as query terms, and the top 100K terms as document terms, resulting in 10^9 pairs. Inputs to the PARM model are at the term level, thus their scores are computed much faster than those of the full cross-encoder that gets long sequences of entire query-document pairs. Only candidate term pairs with scores above 0.1 are included in the relevance thesaurus, resulting in a total of 553,864 term pairs.

4.2 Evaluations

We evaluate the BM25T model by exploiting our built relevance thesaurus in two ways: ranking effectiveness and fidelity. Ranking effectiveness is measured by standard ranking evaluation metrics that use ground truth judgments. It demonstrates to what extent BM25T can be used for relevance ranking. Fidelity expresses the extent to which the BM25T faithfully explains the behavior of the target ranking model, i.e., the cross-encoder model.

To demonstrate the generalizability of the relevance thesaurus obtained from PaRM, we developed QLT (Query Likelihood with Thesaurus), a variant of BM25T (PaRM) based on the query-likelihood (QL) framework (Ponte and Croft, 1998). QLT incorporates the translation language model (Berger and Lafferty, 1999) and uses translation probabilities extracted from our relevance thesaurus. Unlike BM25T which computes the score of a query term based on the most relevant document term, QLT computes the score by summing the relevance scores of the document terms.

To provide a baseline comparison and demonstrate the effectiveness of PaRM in building a relevance thesaurus, we re-purposed a local expla-

[2] cross-encoder/ms-marco-MiniLM-L-6-v2 from https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

[3] https://lucene.apache.org/

Model	TREC DL19	TREC DL20	Dev
	NDCG@10	NDCG@10	MRR
BM25	0.516	0.503	0.160
BM25T (L to G)	0.518	0.501	0.158
BM25T (PaRM)	0.550‡	0.546‡	0.180‡
QL	0.495	0.509	0.153
QLT (PaRM)	0.543‡	0.540‡	0.170‡
Cross-encoder	0.763	0.739	0.375

Table 2: Ranking performance on the MS MARCO driven datasets. ‡ marks the statistically significant difference ($p < 0.01$) to the baseline in each group.

nation method (Llordes et al., 2023) as a global explanation (Lundberg et al., 2020), denoted as BM25T (L to G). Given a query and ranked candidates documents for it, this explanation method identifies which terms in the document are relevant to the query.

We adapted it by aligning each document term to the most relevant query term using a cross-encoder and aggregating alignments across 400K training queries. The aggregated scores of term pairs create a relevance thesaurus, which augments the BM25 scoring function as in BM25T (PaRM) (See Appendix B for details).

In-domain ranking effectiveness. First, we evaluate ranking effectiveness on three datasets derived from MS MARCO. TREC DL 2019 and 2020 (Craswell et al., 2020, 2021) contain 43 and 53 queries, respectively. Top-ranked documents are thoroughly judged by NIST assessors, which make them more reliable for evaluating the ranking effectiveness.

We also used a larger dataset called MS MARCO-dev, which we built by sampling 1,000 queries from the development split of MS MARCO. As this dataset is sparsely judged, with most queries having only one relevant document, we evaluated it using mean reciprocal rank (MRR). MS MARCO-dev will also be used for evaluating fidelity, where more data points are preferable.

Table 2 shows the ranking effectiveness of methods on the MS MARCO datasets. BM25T with our proposed PaRM shows significant gains ($p < 0.01$) over BM25 in all datasets. The obtained gains demonstrate that the distilled relevance thesaurus effectively improves the vocabulary mismatch problem of BM25. In contrast, the BM25T (L to G) does not show consistent improvements. BM25T still has a gap from the cross-encoder model, showing room for improvement in future work. Note that we do not include other retrieval models to compare with their ranking effectiveness, as they cannot be used to make a global explanation. QLT (PaRM) also has better effectiveness compared to QL. Considering that the thesaurus is only tuned for BM25 but not for QL, this result demonstrates the generalization ability of the acquired relevance thesaurus.

Dataset	BM25	BM25T	Cross Encoder
HotpotQA	0.633	0.641†	0.725
DBPedia	0.325	0.350†	0.447
NQ	0.307	0.332†	0.462
Touché-2020	0.499†	0.337	0.272
SCIDOCS	0.150	0.148	0.163
TREC-COVID	0.583	0.602	0.733
FiQA-2018	0.245	0.248	0.341
Quora	0.775†	0.738	0.823
ArguAna	0.407†	0.359	0.311
SciFact	0.678	0.678	0.688
NFCorpus	0.319	0.348†	0.369
ViHealthQA	0.217†	0.173	0.168

Table 3: The ranking effectiveness measure (NDCG@10) of the methods on BEIR datasets. † marks the statistically significant difference ($p < 0.05$) between BM25 and BM25T

Out-of-domain ranking effectiveness. The BEIR benchmark (Thakur et al., 2021) is a collection of IR datasets and is widely used to measure the generalizability of models without domain-specific training. We evaluate the zero-shot ranking effectiveness of the BM25T model over this benchmark, using the same relevance thesaurus distilled from the cross-encoder model that is trained on MS MARCO.

Table 3 shows evaluation results on the BEIR datasets. Out of the 12 datasets, the performance difference between BM25 and BM25T is statistically significant ($p < 0.05$) in 8 datasets. Among these datasets, BM25T outperforms BM25 on 7 datasets, showing a performance closer to that of the cross-encoder. Thus, we conclude that the relevance thesaurus is not limited to the corpus on which it is trained and can effectively perform semantic matching in out-of-domain datasets.

Fidelity. As our task is a ranking task, we measure faithfulness in terms of the correlation between the scores from the explanations and the target model.

Given a query q and its corresponding candidate documents $\{d_1, d_2, ..., d_n\}$, the fidelity of an explanation is computed as the correlation between the scores $\{S_b(q, d_1), S_b(q, d_2), ..., S_b(q, d_n)\}$ from the targeted neural model and the global predictor from explanations (e.g., BM25T) $\{S_e(q, d_1), S_e(q, d_2), ..., S_e(q, d_n)\}$.

For the fidelity score for a dataset, we calculate this correlation for each query and then average them across all queries in the dataset. For each query, the top 1,000 documents retrieved by BM25 are used as candidate documents. The Pearson correlation coefficient, which ranges from -1 to 1, is used as a measure of correlation, with 1 indicating the strongest positive correlation. Appendix C reports results for other correlation measures, which show similar findings.

In addition to the cross-encoder model, which was used for training PaRM, we apply fidelity evaluation to the other IR models that fine-tune Transformer-based models on MS MARCO. Four popular retrieval models are included. The first two models are TAS-B (Hofstätter et al., 2021) and Splade v2 (Formal et al., 2021a) which are trained using knowledge distillation from the cross-encoders. The next two are Contriever and Contriever+M (Izacard et al., 2021). Contriever is trained with unsupervised learning unsupervised, and Contriever+M is the model that further finetunes Contriever using MS MARCO. These four models are dual-encoders, where the query and document are independently encoded into vectors using Transformer encoders.

	Fidelity		Ranking
Ranking Model	BM25	BM25T	MRR
Cross-encoder	0.484	0.580	0.375
Splade v2	0.490	0.583	0.335
TAS-B	0.421	0.513	0.318
Contriever	0.417	0.454	0.174
Contriever+M	0.411	0.495	0.307

Table 4: Fidelity of the explanations to the ranking models, measured by Pearson correlations on the MS MARCO Dev dataset. Both BM25 and BM25T are considered explanations for the corresponding ranking models. The ranking performance, measured by Mean Reciprocal Rank (MRR), is provided as a reference.

Table 4 shows the fidelity of BM25 and BM25T to these neural retrieval models on MS MARCO-dev. First, we can observe that in all ranking models, BM25T has higher fidelity than BM25. Also, the gain is larger on models that are trained on MS MARCO. Contriever is not trained based on the MS MARCO dataset, on which BM25T showed the lowest fidelity and smallest fidelity gain.

We conclude that the relevance thesaurus can serve to explain the behavior of the models that are trained with similar training data.

The fidelity evaluations on BEIR datasets also confirm that BM25T is more faithful than BM25 in explaining the cross-encoder model (Table 10). The fidelity of BM25T is higher than BM25 across all datasets, except in the Quora dataset. The average fidelity across the datasets improved from 0.507 with BM25 to 0.630 with BM25T.

The high fidelity of BM25T to the cross-encoder model is further evidenced by its performance across the BEIR datasets, as shown in Table 3. In fact, BM25T mirrors the CE's performance drops in the ArguAna and Touche-2020 datasets. This consistency suggests that the relevance thesaurus effectively captures the semantic matching patterns of the CE, even when those patterns lead to decreased performance. Further analysis of the relevance thesaurus could provide insights into why the additional semantic matches sometimes result in worse performance in these specific datasets

4.3 Insights from the relevance thesaurus

The relevance thesaurus contains both reasonable and unexpected term pairs, as illustrated in Table 1. Through the analysis of the thesaurus, we identified three interesting findings.

Car-brand bias The thesaurus reveals that the models associate "car" with many brand names, but assign higher scores to certain brands over others (Figure 4). For example, the pair ("car", "Ford") has a score of 0.39, while ("car", "Honda") has a score of 0.28.

This bias can impact the ranking of documents in the following way: imagine a query containing the term "car" and two documents that are identical except for the mentioned car brand - one document includes a high-scoring brand, while the other features a low-scoring brand. Due to the higher score assigned by the thesaurus, BM25T will rank the document with the high-scoring brand above the one with the low-scoring brand. This observation suggests that the neural ranking model is likely to exhibit the same bias, prioritizing documents that mention high-scoring brands over those with low-scoring brands, even when the documents' content is otherwise the same.

When-year bias The models exhibit a temporal bias, assigning different scores to various years for the query term "when", with much lower scores

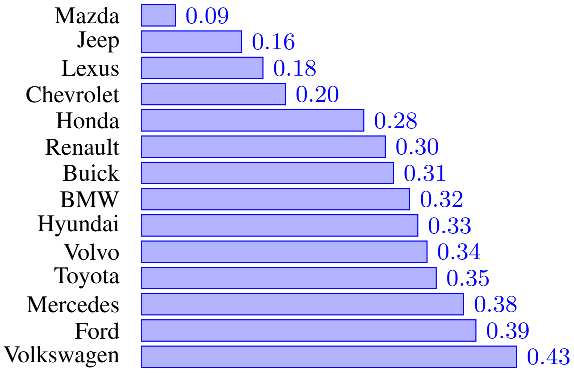


Figure 4: Scores for car brand names against the query term "car" based on our relevance thesaurus.

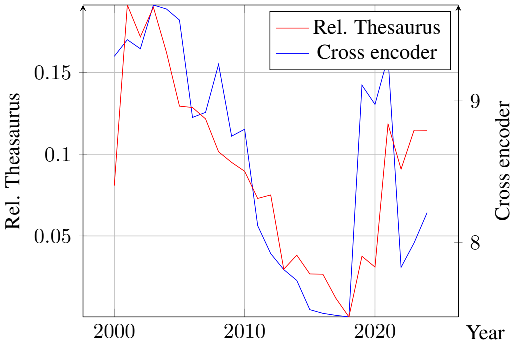


Figure 5: Relevance scores for the query term "when" and document terms representing years from 2000 to 2024, based on our relevance thesaurus and the cross-encoder model.

for years around 2015, when the MS MARCO dataset was constructed (Figure 5). We hypothesize this bias exists because the current year is often less informative for "when" questions, as more specific temporal information is typically expected. While effective for 2015 data, it could lead to sub-optimal performance for different current years, such as 2024.

Experiments detailed in Appendix A validate the presence of these behaviors in state-of-the-art relevance models, supporting that the behaviors specified by the relevance thesaurus are well representing the neural ranking models.

Postfix a Many thesaurus entries consist of cases where the document term is the query term with an additional "a" or "â" at the end, such as the ("car", "vehicleâ"). This is due to encoding errors in the MS MARCO dataset, where the right quotation marks (') were incorrectly decoded as "â". The issue is compounded by the BERT tokenizer's normalization of "â" to "a". For example, the system might erroneously consider "cud" (partially digested food in a cow's stomach) relevant to "CUDA" (NVIDIA's parallel computing platform).

5 Conclusion

We explored using a relevance thesaurus as a global explanation for neural ranking models. We proposed an effective approach for constructing the thesaurus by training a partial relevance model (PaRM). Augmenting the acquired thesaurus into BM25 enhanced its ranking effectiveness and fidelity to the targeted neural ranking model across multiple information retrieval datasets. Furthermore, the thesaurus uncovered unexpected corpus-specific behaviors and biases of state-of-the-art ranking models, highlighting its value in identifying potential issues and limitations in neural rankers.

We expect a few promising research directions on top of our work. The proposed strategy of using thesaurus to explain a model can be further extended to other Transformer-based models, including generative language models to discover biases on these models. For IR applications, effectiveness of BM25T could be improved by considering multiple document terms for each query term and incorporating term location information. These enhancements would better mimic neural ranking models' behavior, potentially leading to more efficient and interpretable sparse retrieval models that more closely match the performance of their neural counterparts.

Acknowledgement

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #2106282. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

Limitations

Our explanation methods have shown the existence of bias in models trained on the MS MARCO dataset. However, the presence of bias does not necessarily indicate inappropriate behavior. In some cases, bias may actually contribute to effective ranking, as certain keywords have a higher likelihood of being relevant due to their ability to be used in different contexts, such as referring to a person or other entities.

While our term replacement experiments were designed to control for context by maintaining identical conditions, real-world documents often exhibit diverse contexts that could potentially diminish the impact of biases. In practice, the contextual differences between documents may result in greater variations in relevance scores compared to the variations caused by biases alone. Consequently, the biases observed in our controlled experiments may have a less significant effect on the ranking of real documents, as the influence of context differences could be more dominant.

The biases identified from the specific relevance thesaurus in our experiments are limited to the ranking models trained on the MS MARCO dataset by fine-tuning BERT-based models. This work has not covered ablations to determine if these biases originated from MS MARCO training data or from language model pre-training of BERT.

While our experiments demonstrated that training PaRM is effective with MS MARCO data, this approach may not be equally effective in low-resource settings. Specifically, the proposed distillation steps require well-representative queries for the datasets.

References

Adam Berger and John Lafferty. 1999. Information retrieval as statistical translation. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229.

Leonid Boytsov and Zico Kolter. 2021. Exploring classic and neural lexical translation models for information retrieval: Interpretability, effectiveness, and efficiency benefits. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*, pages 63–78. Springer.

Olcay Boz. 2002. Extracting decision trees from trained neural networks. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 456–461.

Tanya Chowdhury, Razieh Rahimi, and James Allan. 2023. Rank-lime: local model-agnostic feature attribution for learning to rank. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 33–37.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track.

Mark Craven and Jude Shavlik. 1995. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*, 8.

W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021b. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42.

Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. 2021. Explaining explanations: Axiomatic feature interactions for deep networks. *Journal of Machine Learning Research*, 22(104):1–54.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48.

Youngwoo Kim, Myungha Jang, and James Allan. 2020. Explaining text matching on neural natural language inference. *ACM Transactions on Information Systems (TOIS)*, 38(4):1–23.

Youngwoo Kim, Razieh Rahimi, and James Allan. 2022. Alignment rationale for query-document relevance. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2489–2494.

Youngwoo Kim, Razieh Rahimi, and James Allan. 2023. Conditional natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6833–6851, Singapore. Association for Computational Linguistics.

Youngwoo Kim, Razieh Rahimi, Hamed Bonab, and James Allan. 2021. Query-driven segment selection for ranking long documents. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3147–3151.

Robert Krovetz. 1993. Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–202.

Jimmy Lin. 2021. Github/castorini/anserini - clean up garbage characters in ms marco dataset.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.

Michael Llordes, Debasis Ganguly, Sumit Bhatia, and Chirag Agarwal. 2023. Explain like i am bm25: Interpreting a dense model's ranked-list with a sparse approximation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 1976–1980, New York, NY, USA. Association for Computing Machinery.

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67.

Lijun Lyu and Avishek Anand. 2023. Listwise explanations for ranking models using multiple explainers. In *European Conference on Information Retrieval*, pages 653–668. Springer.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 1101–1104.

Chandler May, Alex Wang, Shikha Bordia, Samuel Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628.

Shahrzad Naseri, Jeffrey Dalton, Andrew Yates, and James Allan. 2021. Ceqe: Contextualized embeddings for query expansion. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*, pages 467–482. Springer.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@NIPS*.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint*, 6:2.

Saran Pandian, Debasis Ganguly, and Sean MacAvaney. 2024. Evaluating the explainability of neural rankers. In *European Conference on Information Retrieval*, pages 369–383. Springer.

Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Dheeraj Rajagopal, Vidhisha Balachandran, Eduard H Hovy, and Yulia Tsvetkov. 2021. Selfexplain: A self-explaining architecture for neural text classifiers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 836–850.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Joe Stacey, Pasquale Minervini, Haim Dubossarsky, and Marek Rei. 2022. Logical reasoning with span predictions: Span-level logical atoms for interpretable and robust nli models. *arXiv preprint arXiv:2205.11432*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran.

Manisha Verma and Debasis Ganguly. 2019. Lirme: Locally interpretable ranking model explanation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1281–1284.

Zijun Wu, Atharva Naik, Zi Xuan Zhang, and Lili Mou. 2021. Weakly supervised explainable phrasal reasoning with neural fuzzy logic. *arXiv preprint arXiv:2109.08927*.

A Unexpected behaviors

In subsection 4.3, we discovered three unexpected behaviors from the relevance thesaurus. This part describes the experiments that support the existence of the behaviors found in the state-of-the-art relevance models. The three behaviors investigated are (1) Postfix a, where the models treat “a” at the end of a word as a quotation mark or apostrophe due to encoding errors in the MS MARCO dataset; (2) Car-brand bias, where the models assign higher scores to certain car brand names over others; and (3) When-year bias, where the models exhibit temporal bias in assigning relevance scores to different years for queries containing the term “when”.

A.1 Car-brand bias

Brand name	Scores	Brand name	Scores
Volkswagen	0.429	Buick	0.308
Ferrari	0.410	Cadillac	0.303
Porsche	0.405	Renault	0.300
Fiat	0.394	Honda	0.279
Chrysler	0.390	Audi	0.269
Ford	0.389	Peugeot	0.269
Mercedes	0.377	Pontiac	0.259
Packard	0.366	Daimler	0.219
Oldsmobile	0.365	Mitsubishi	0.212
Toyota	0.350	Nissan	0.205
Jaguar	0.348	Chevrolet	0.202
Volvo	0.341	Lexus	0.180
Hyundai	0.332	Jeep	0.159
BMW	0.324	Mazda	0.094
Bentley	0.322		

Table 5: Scores for each of 29 brand names against the query term “car” based on our relevance thesaurus.

The relevance thesaurus reveals that the models associate the query term “car” with many brand names, such as “Ford” and “Honda”, but consistently assign higher scores to certain brand names over others (Table 5). To verify if this bias is present in the state-of-the-art relevance ranking models, we designed an experiment using the MS MARCO passage collection.

From the training split of the MS MARCO passage collection, we selected queries that include the term “car” but exclude any car brand names or content specific to particular brands. We then selected documents for each of the queries that satisfy the following criteria:

1. The document mentions only one brand name.

2. The document does not contain any brand-specific information when the brand name is removed.
3. The document is predicted as relevant by the cross-encoder model.

To filter the documents based on the second and third criteria, we employed keyword-based filtering using a list of car models, ChatGPT-based filtering to identify brand-specific information and manual annotations. For the keyword filtering, we built a list of car models and excluded the documents that contained any of the model names. For ChatGPT-based filtering, we masked the brand name mention of the document and prompted, "Does this document contain any brand-specific information?", and if the answer was yes, the document was excluded. This process resulted in 382 query-document pairs, with 29 car brand names considered.

For each query-document pair, the brand name mentioned in the document was replaced by each of the 29 brand names in turn. All the resulting combinations were scored by neural ranking models, yielding a score array of 382 × 29, where each row represents a query-document pair and each column represents a brand name. In other words, the element at position (i, j) in the array represents the score assigned by the neural ranking model to the i-th query-document pair when the brand name is replaced by the j-th brand name. The element (i, i) represents the original brand that appeared in the document.

To obtain a single score per brand name, we averaged the scores across the 382 query-document pairs. We then measured the correlation between these average scores and the scores from the relevance table derived from the cross-encoder model. If a neural model exhibits the bias suggested by the representation, we expect a corresponding bias in the modified documents.

Table 6 shows the correlation values (fidelity) obtained for each of the ranking models. The results demonstrate that the scores from the thesaurus correlate with scores from the neural ranking models, indicating that our relevance thesaurus can be used to identify possible biases of ranking models.

A.2 When-year

The models exhibit a temporal bias where different years have different scores for the query term "when", with much lower scores assigned to the years around 2015 compared to other years. Most years (e.g., "2001") have high relevance scores for the query term "when" in the relevance thesaurus, but the score sharply decreased around 2015, the year when the MS MARCO dataset was constructed (Figure 5).

Ranking Model	Car - brand	When - year
Cross Encoder	0.282	0.746
Splade v2	0.413	0.224
TAS-B	0.367	0.484
Contriever	0.419	0.422
Contriever + M	0.200	0.665

Table 6: The fidelity of relevance thesaurus focused on two findings. The models predict scores on query-document pairs when a brand name or year mention is replaced with another.

To validate if this bias is in neural models, We measured the predicted scores from the neural ranking models with the query being "when did North Carolina join IFTA" and the document being "year North Carolina join IFTA". Table 6 shows the correlation between the scores from the ranking model and the relevance thesaurus.

The results show that the neural ranking models exhibit a similar temporal bias to the relevance thesaurus, where documents mentioning years around 2016 are scored lower in relevance for queries containing the term "when". This correlation confirms that our relevance thesaurus faithfully captures the biases underlying the neural models.

A.3 Postfix A

Many entries in the thesaurus consist of cases where the document term is the query term with an additional "a" or "â" at the end, such as ("car", "vehicleâ"). This is due to encoding errors in the MS MARCO dataset, where the right quotation marks (') were incorrectly decoded as "â". When combined with the BERT tokenizer normalizing "â" to "a", it could result in incorrect matching, such as considering the term "cud" (food in cows' stomach) to be relevant to "CUDA" (parallel computing platform).

To test the postfix hypothesis, we selected 500 relevant query-document pairs that contained a common term. We then appended characters from "a" to "z" to the document occurrence of this common term. For each appended character, we mea-

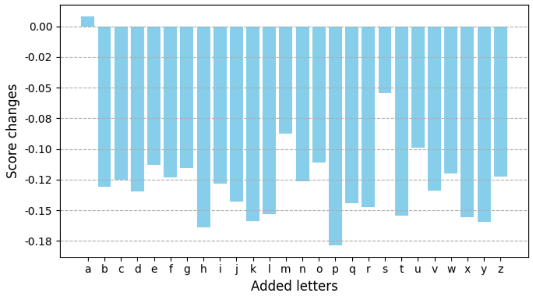


Figure 6: Postfix-a experiment results for the cross encoder. Modifying the matching document term by appending any alphabet results in a large score drop, except ‘a’.

sured the change in the relevance score assigned by the model.

The result on the cross encoder as illustrated in Figure 6 shows that while all other alphabet characters result in a large score drop when they are appended to the query term occurrence in the document, appending “a” actually results in a small increase of the relevance score. The difference between score changes of “a” and other cases are all statistically significant at $p < 0.01$. This supports the existence of many entries in the relevance thesaurus where the document term has the additional character “a” at the end of the terms.

The results, illustrated in Figure 6, show that appending “a” leads to a small increase in the relevance score, while all other characters result in a significant score drop ($p < 0.01$). This supports the existence of entries in the relevance thesaurus where the document term has an additional “a” at the end. Similar behaviors were observed on Splade and Contriever-MS MARCO, while Contriever and TAS-B do not exhibit such behavior (Table 7).

While the existence of encoding errors is known (Lin, 2021), there has been no systematic analysis of how these errors could affect the ranking models. This analysis shows that our thesaurus explanation can be effectively used to discover that the model is using features that may not generalize to other corpora.

B Experiments details

B.1 BM25T with Local to Global

As one of our baselines, we adapted the local explanation method proposed by Llordes et al. (2023) into a global explanation. Their approach provides local explanations for a given query and ranked documents by identifying matching document terms. The local explanation method can be represented as a function:

$$E : (q, d_1, d_2, ..., d_n) \rightarrow (w_1, w_2, w_3, ...)$$

, where q is a query, d_i is a ranked document, and w_i is a document term that is considered to match the query.

While it has been argued that local explanations can be converted into global explanations by accumulating them (Janizek et al., 2021), the local explanations for IR models often lack the necessary information to be used globally. Specifically, the document terms (w_i) are not attributed to specific query terms, which is a crucial requirement for building a term-level global thesaurus.

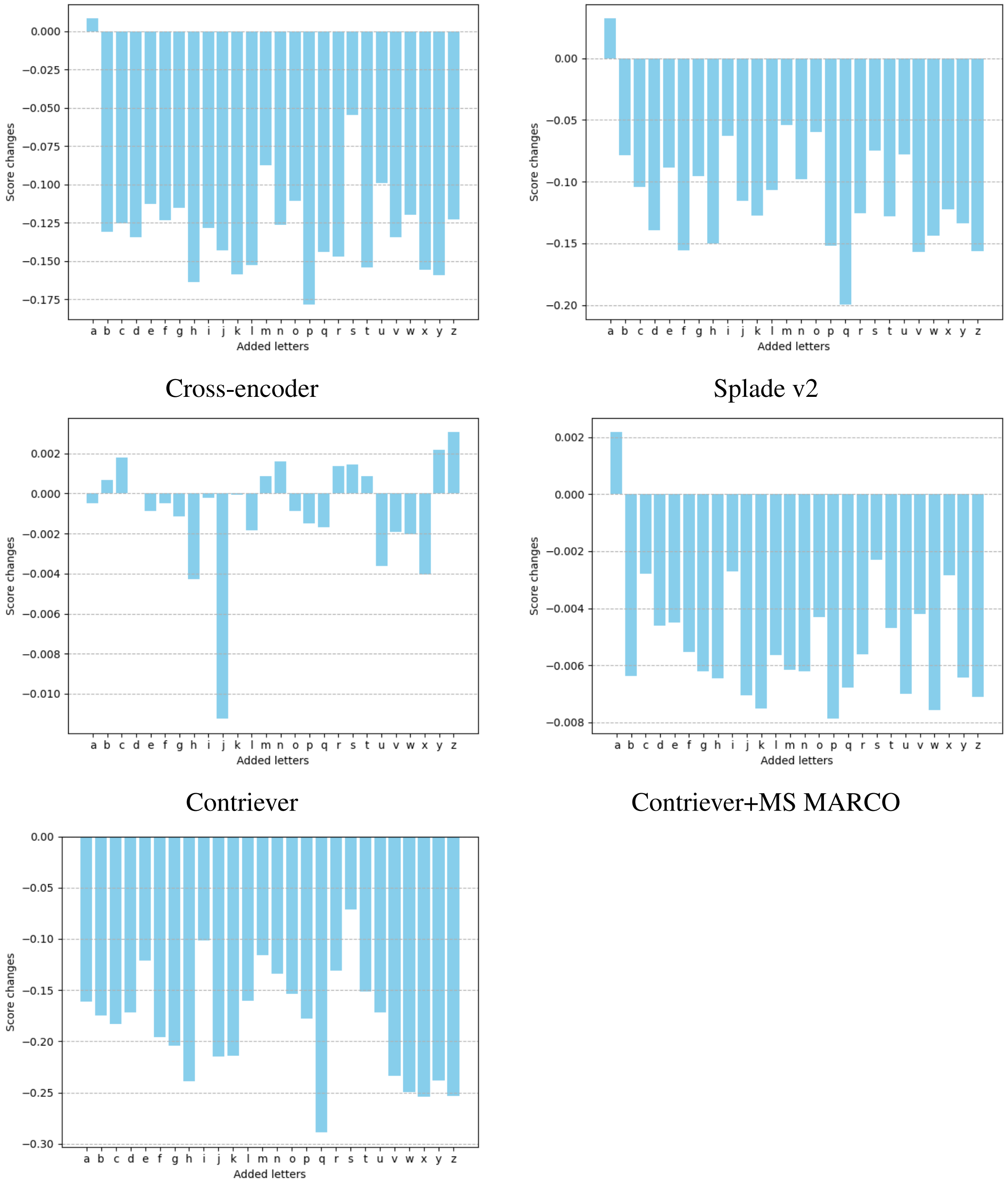
To build a global explanation in the form of a relevance thesaurus, we align each document term to one of the query terms using the following method. Using a cross-encoder ranker, we compute relevance scores between each query term and each document term, considering them as standalone queries and documents, respectively. The query term with the highest score is aligned to the corresponding document term.

We apply this alignment process to 400K training queries and their re-ranked candidate documents using the cross-encoder. Compared to PaRM, this method uses a similar number of queries but a larger number of documents per query. For each query, 1,000 candidate documents were re-ranked and the top 10 documents were used to select the document terms for explanation building.

The score between a query term qt and a document term dt is calculated as the number of times qt is aligned to dt divided by the total number of occurrences of qt in the queries. We only included the document terms which appear more than once, to reduce the noise. Table 8 shows the result for the different variants for BM25T (L to G).

B.2 QLT

We implemented and evaluated the performance of QLT (Query Likelihood with Thesaurus) to test



Cross-encoder

Splade v2

Contriever

Contriever+MS MARCO

TAS-B

Table 7: Postfix-a experiments results. The listed scores are average of (score after change – score before change), and positive values indicate score increases and negative values indicate score decreases.

	TREC DL 19	TREC DL 20
BM25	0.516	0.503
BM25T		
PaRM	0.550	0.546
L to G	0.504	0.501
+ Min frequency filter	0.518	0.501
+ Uniform attribution	0.475	0.493

Table 8: The ranking effectiveness of BM25T (L to G) with different configurations

how the relevance thesaurus can be used for different models that the thesaurus was not optimized for. One key difference between QLT and BM25T is that QLT computes the score as the sum of the relevance scores of the document terms, while BM25T computes the query term's score based on the most relevant document term.

In the original query likelihood model, the relevance score for query q and document d is computed as:

$$p(\mathbf{q}|\mathbf{d}) = \prod_i p(q_i|\mathbf{d}), \tag{8}$$

where $p(q_i|\mathbf{d})$ is the probability of the query term q_i in the document d, calculated as the frequency of q_i in d divided by the length of d.

The translation language model for information retrieval (Berger and Lafferty, 1999) considers that any document term w can be ”translated” into the query term q_i with the translation probability $t(q_i|w)$. The term probability is then computed as:

$$p(q_i|\mathbf{d}) = \sum_w t(q_i|w)p(w|\mathbf{d}), \tag{9}$$

where $t(q_j|w)$ is the translation probability, and $p(w|\mathbf{d})$ is the probability of the word w in the document d, again computed as the frequency of q_i in d divided by the length of d. In QLT, we adopt the translation language model while using our relevance thesaurus to compute the translation probability.

B.3 BM25 and Query Likelihood (QL) configuration

For the hyper-parameters of BM25, we used the default values ($k_1 = 0.9$ and $b = 0.4$) as in Pyserini (Lin et al., 2021). We used the analyzer configurations as in the Pyserini implementation, which includes normalization, tokenization, stemming, and stopwords removal.

Our query likelihood implementation uses the same tokenizer as in BM25. We used Dirichlet smoothing (Croft et al., 2010), and tuned the parameter μ on another validation set, as the default values were ineffective for short passages.

C Full experiments results

C.1 Different fidelity metrics

Many works have used different metrics for the fidelity evaluation of ranked lists in IR tasks, such as the overlap of **top-k** in the ranked list (Llordes et al., 2023), agreement rates of **pairwise** preference agreement in the ranked list (Lyu and Anand, 2023), or **Kendall** rank correlations (Pandian et al., 2024).

We chose the Pearson correlation as the main metric because it considers the magnitude of score differences rather than just the rankings. This is particularly important in IR tasks, where the emphasis is on differentiating a few highly relevant documents from the many non-relevant ones.

While IR tasks are indeed ranking tasks, they prioritize top-ranked documents, which are more likely to be relevant. As a result, ranking correlations like the Kendall rank correlation may be less appropriate, as they are more affected by the ranking of non-relevant documents, which outnumber relevant ones. We also consider the overlap rates of top-k to be less desirable for two reasons: first, the number of relevant documents is unknown; second, it does not account for score differences among the top-k items, which are crucial for reliable IR metrics like NDCG. Nevertheless, we have included results for other fidelity metrics (Table 9), which consistently confirm the improvements of BM25T (PaRM) over BM25.

C.2 Fidelity on BEIR

In subsection 4.2, we include only the ranking effectiveness of BM25T on the BEIR dataset, and not the fidelity (correlation). Table 10 shows the correlation of the scores when BM25 or BM25T is considered as an explanation and compared to scores from the cross-encoder model. In most datasets, the correlations increased, with the only exception of the Quora dataset.

The improvements in ranking performance are consistent with the increased correlations observed in most datasets.

D PaRM implementation details

In the first phase, PaRM calculates a relevance score for the given query-document by generating

	Pearson r		Kendall τ		Pairwise		Top-k overlap	
Model	BM25	BM25T	BM25	BM25T	BM25	BM25T	BM25	BM25T
Cross Encoder	0.484	0.580	0.260	0.341	0.632	0.672	0.293	0.334
Splade v2	0.490	0.583	0.268	0.346	0.636	0.674	0.304	0.345
TAS-B	0.421	0.513	0.228	0.304	0.616	0.653	0.256	0.293
Contriever	0.417	0.454	0.230	0.259	0.617	0.631	0.273	0.289
Contriever + M	0.411	0.495	0.225	0.294	0.615	0.648	0.264	0.303

Table 9: Fidelity of BM25 and BM25T when measured by different metrics.

Dataset	BM25	BM25T
HotpotQA	0.535	0.647
DBPedia	0.477	0.612
NQ	0.474	0.658
Touché-2020	0.403	0.689
SCIDOCS	0.598	0.663
TREC-COVID	0.276	0.705
FiQA-2018	0.481	0.514
Quora	0.659	0.640
ArguAna	0.656	0.722
SciFact	0.634	0.677
NFCorpus	0.584	0.626
ViHealthQA	0.314	0.410

Table 10: Fidelity (Pearson correlation) of the BM25 and BM25T as explanation to the cross encoder ranking model.

scores for two inputs, (q_1, d_1) and (q_2, d_2), which are built from the given query q and document d. Each of the two inputs is then scored through PaRM and these two scores are summed as the relevance score for the query-document pair. Using the relevance label for the query and document, PaRM is trained end-to-end to predict relevance for partial sequences of the query and document without fine-grained labels.

We build q_1 by extracting a continuous span from q and build q_2 with the remaining tokens, leaving a [MASK] token where q_1 was extracted. Given q_1, q_2, and d, we build d_1 and d_2 by masking some tokens of the document d, while keeping tokens that are likely to be relevant to the corresponding q_i. Both d_1 and d_2 can be composed of many non-continuous spans.

D.1 Building partial segments for the first phase

The tokens to be deleted are selected so that the deleted tokens in d_i are less likely to be important for the corresponding query partition q_i. To estimate which tokens of the document are less likely to be important for a query partition, we use the attention scores from a canonical cross-encoder model which takes the concatenation of whole query q and document d as an input.

The scoring is done in the following steps.

1. We collect normalized attention probabilities from all the layers and heads of the Transformer network. As a result, we get a four dimension tensor of $W \in \mathbb{R}^{L\times L\times M\times H}$, where L is the sequence length, M is the number of layers, and H is the number of attention heads in each layer. W_{ijlk} denote the attention probability for the i-th token to attend to the j-th token in the k-th attention head of the l-th layer.

2. We average W over the last two dimensions, which correspond to different layers and heads, and get a two-dimensional matrix A.
$$A_{ij} = \sum_l \sum_k W_{ijlk} \tag{10}$$

3. Let $|q|$ be the number of tokens in the query and $|d|$ to be the number of tokens in the document. When a [CLS] token and [SEP] tokens are combined with the query and document tokens, the query tokens are located from the second token to $(|q| + 1)$-th token, and the document tokens are located from $(|q| + 3)$-th token to $(|q| + |d| + 2)$-th.

   Then, $A_{2:|q|+1,|q|+3:|q|+|d|+2}$ indicates the averaged probability that query tokens to attend to document tokens, and $A_{|q|+3:|q|+|d|+2,2:|q|+1}$ indicates averaged probability that document tokens attend to query tokens.

4. By transposing the latter matrix and adding it to the first, we obtain S. In this resultant

matrix, S_{ij} indicates the degree of attention between the i-th token of the query and j-th token of the document.

To avoid splitting a word into subwords, the token selection was performed at the word-token level instead of the subword-token. The subword-token-level scores are converted by taking maximum scores.

We select the tokens with the lowest attention probability for the corresponding query partition as ones to be deleted.

The number of tokens to be deleted from the document is randomly sampled so that it can have variable inputs starting from a single token to nearly a full sequence. The number of deleted tokens m is sampled from a normal distribution with the mean and standard deviation being half of the document length $|d|$. The sampled number is capped at a minimum of 1 and a maximum of $|d| - 1$.

D.2 Second Phase

In the second phase, PaRM is trained on word pairs, where the query term does not appear in the document. Here are a few clarifications about the details.

To select term-pair candidates, we pre-computed scores for the term pairs by limiting them to the frequent terms. Similar to the relevance thesaurus itself, the top 10K frequent terms were considered as query terms, and the top 100K terms as document terms. The terms without scores are not selected for training.

The terms that are fed to PaRm are stemmed in the same way they are used for BM25T. However, stopwords are NOT excluded for this step, while BM25T excludes them.

D.3 Training configurations

Both phases use the query-documents triplets provided with MS MARCO passage ranking dataset (Nguyen et al., 2016), which makes about 400,000 training instances.

For the first stage training of PaRM, we apply early stopping based on the loss validation set. We used the batch size of 16 and learning rate of 2e-5, which were not tuned.

For the second stage of training of PaRM, we tuned the learning rate and batch sizes based on its loss on holdout split and BM25T augmented performance (MRR) in a validation set that is separately sampled from MS MARCO dev. The reported model used a batch size of 256 and a learning rate of 1e-5.

D.4 Computational cost

We used four NVIDIA GeForce GTX 1080 Ti GPUs for training. Both the first and second stages of training took less than 10 hours each. The inference for relevance thesaurus construction was run on approximately 100 GPUs, including GeForce GTX 1080 Ti and GTX Titan X models, and took about 700 GPU hours.

E Responsibility statement

E.1 Artifact - MS MARCO Dataset

The MS MARCO dataset (Nguyen et al., 2016), used as the artifact in this paper, has been carefully curated and anonymized by its creators to protect user privacy and prevent the inclusion of personally identifying information or offensive content. The dataset consists of anonymized search queries and corresponding relevant passages from web pages, processed to remove any personal information.

The MS MARCO dataset is a large-scale information retrieval dataset covering a wide range of domains and topics in the English language. It includes real-world search queries from Bing and corresponding relevant passages from web pages. The dataset is divided into training, development, and testing sets [4], each containing a substantial number of query-passage pairs. While demographic information is not explicitly provided due to privacy concerns, the dataset is considered representative of diverse information needs and user intents in web search scenarios.

E.2 AI Assitance

We acknowledge the use of AI assistants, Claude by Anthropic [5] and GPT-4 by OpenAI [6], in the writing process of this paper. These AI assistants provided support in drafting and refining the contents of the paper. However, all final decisions regarding the content, structure, and claims were made by the human authors, who carefully reviewed and edited the generated content.

[4] https://microsoft.github.io/msmarco/Datasets

[5] https://www.anthropic.com/claude

[6] https://chat.openai.com/