Title: DeepImpact: Enhancing Document Retrieval with Inverted Indexes

DeepImpact: Enhancing Document Retrieval with Inverted Indexes

In recent years, there have been significant advancements in the field of document retrieval, particularly in the development of models that can compute conditional probabilities for improved retrieval performance [REF0]. One such approach is DeepImpact, which combines traditional inverted indexes with contextualized language models to enhance document retrieval [REF5]. By estimating the semantic importance of tokens in a document collection, DeepImpact generates impact scores that aid in efficient retrieval.

The use of inverted indexes is a well-established technique in information retrieval systems. It involves creating an index that maps terms to the documents in which they appear, allowing for fast and efficient retrieval of relevant documents [REF3]. However, traditional inverted indexes do not take into account the semantic meaning of the terms, which can limit their effectiveness in capturing the relevance of documents to a given query.

DeepImpact addresses this limitation by leveraging contextualized language models, such as the Transformer architecture, to estimate the semantic importance of tokens in a document collection [REF0]. By considering both the input query and the task to be performed, DeepImpact models the conditional distribution of the output given the input and task, i.e., p(output|input, task) [REF0]. This approach allows for more accurate retrieval of relevant documents, as it takes into account the specific task at hand.

To evaluate the effectiveness of DeepImpact, several experiments have been conducted using benchmark datasets such as MS MARCO passage ranking [REF6]. These experiments compare DeepImpact with other retrieval methods, including traditional inverted indexes and state-of-the-art neural retrieval approaches [REF3]. The results show that DeepImpact outperforms the inverted-index based baselines and even matches the effectiveness of more complex neural retrieval approaches like ColBERT [REF5].

In addition to its improved retrieval performance, DeepImpact also offers benefits in terms of query processing latency. By using approximate nearest neighbor techniques, DeepImpact reduces the time required for query processing, making it a more efficient solution [REF5]. This reduction in latency is particularly significant when DeepImpact is used in combination with re-ranking methods, such as ColBERT, which further improves the overall effectiveness of the retrieval pipeline [REF5].

Future work on DeepImpact aims to enhance the underlying model by exploring more relaxed matching conditions between query-document terms [REF5]. This could potentially improve the retrieval performance by allowing for a wider range of relevant documents to be retrieved. Additionally, further research is needed to evaluate the robustness of DeepImpact across different domains and tasks, as well as to explore multitask learning frameworks for improving general performance [REF7].

In conclusion, DeepImpact offers a novel approach to document retrieval by enhancing traditional inverted indexes with contextualized language models. Its ability to estimate the semantic importance of tokens in a document collection leads to improved retrieval performance and reduced query processing latency. The experimental results demonstrate the effectiveness of DeepImpact in comparison to other retrieval methods, highlighting its potential for enhancing document retrieval systems [REF8].

References given to GPT:
[REF0] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Ranking_with_Encoder-decoder_Models/BIBREF41_9405cc0d6169988371b2755e573cc28650d14dfe.pdf Title: Language Models are Unsupervised Multask LearnersChunk of text: Yn i=1 p(sn|s1, ..., sn-1) (1) This approach allows for tractable sampling from and es timation of p(x) as well as any conditionals of the form p(sn-k, ..., sn|s1, ..., sn-k-1). In recent years, there have been significant improvements in the expressiveness of mod els that can compute these conditional probabilities, such as self-attention architectures like the Transformer (Vaswani et al., 2017). Learning to perform a single task can be expressed in a probabilistic framework as estimating a conditional distri bution p(output|input). Since a general system should be able to perform many different tasks, even for the same input, it should condition not only on the input but also on the task to be performed. That is, it should model p(output|input, task). This has been variously formalized in multitask and meta-learning settings. Task conditioning is often implemented at an architectural level, such as the task specific encoders and decoders in (Kaiser et al., 2017) or at an algorithmic level such as the inner and outer loop optimization framework of MAML (Finn et al., 2017).

[REF1] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Ranking_with_Encoder-decoder_Models/BIBREF41_9405cc0d6169988371b2755e573cc28650d14dfe.pdf Title: Language Models are Unsupervised Multask LearnersChunk of text: Figure 5. CDF of percentage 8-gram overlap with WebText train ing set, for both WebText test set and samples (conditioned on WebText test set, with top-k truncated random sampling with k = 40). Most samples have less than 1% overlap, including over 30% of samples with no overlap, whereas the median for test set is 2.6% overlap. 8.3. Diversity Table 12 shows multiple completions of the same random WebText test set context, showing the diversity of completions with standard sampling settings. 8.4. Robustness Table 13 shows the previously mentioned talking unicorns news article.

[REF2] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Vector_quantisation_approaches/BIBREF68_4748d22348e72e6e06c2476486afddbc76e5eca7.pdf Title: Product Quantization for Nearest Neighbor Search Chunk of text: (right). The mean squared error on the distance is on average bounded by the quantization error. k * = 256 and m = 8 is often a reasonable choice. III. SEARCHING WITH QUANTIZATION Nearest neighbor search depends on the distances between the query vector and the database vectors, or equivalently

the squared distances. The method intro duced in this section compares the vectors based on their quantization indices, in the spirit of source coding techniques.

[REF3] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Impact_score_learning/BIBREF91_4aa1d28944856ebe1950a27f633c6667ead3cbf8.pdf Title: Learning Passage Impacts for Inverted Indexes
Chunk of text: Baselines. We perform two different sets of experiments. Our ini tial experiment aims at comparing the performance of DeepImpact as a first-stage ranker, processing queries on inverted indexes but without complex reranking. In this experiment we compare our pro posed DeepImpact with the classical BM25 relevance model over the unmodified collection, and state-of-the-art solutions dealing with inverted indexes, namely DeepCT, and BM25 over a collection expanded with DocT5Query. We do not compare with DeepCT over the collection expanded with DocT5Query, since that would involve training a new DeepCT model from scratch to learn how to weigh 1We have made a submission to the official leaderboard and obtained an MRR@10 of 0.318 on the "eval" queries. expanded documents. Our second set of experiments compares DeepImpact in a re-ranking setting.

[REF4] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Vector_quantisation_approaches/BIBREF68_4748d22348e72e6e06c2476486afddbc76e5eca7.pdf Title: Product Quantization for Nearest Neighbor Search Chunk of text: The comparison is focused on large scale indexing, i.e., we do not consider the im pact of a post-verification step , or geometrical 0.4 0.45 0.5 0.55 0.6 0.65 0.7 0.75 0.8 1k 10k 100k 1M mAP database size (number of images) IVF+HE 64 bits IVFADC 64 bits (m=8) IVF+HE 32 bits IVFADC 32 bits (m=4) Fig. 12. Comparison of IVFADC and the Hamming Embedding method of . mAP for the Holidays dataset as function of the number of distractor images (up to 1 million). information . Figure 12 shows the search performance in terms of mean average precision as a function of the size of the dataset. We have used the same coarse quantizer (k '=20,000) and a single assignment strategy (w=1) for both the approaches, and fixed k *=256 for IVFADC.

[REF5] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Impact_score_learning/BIBREF91_4aa1d28944856ebe1950a27f633c6667ead3cbf8.pdf Title: Learning Passage Impacts for Inverted Indexes
Chunk of text: (re-rank stage) 10 0.270∇ 0.302∇ 0.341∇ 0.350 20 0.299∇ 0.322∇ 0.355 0.357 200 0.343∇ 0.353∇ 0.361 0.361 1000 0.355∇ 0.360 0.362 0.362 4 CONCLUSIONS AND FUTURE WORK In this paper, we introduced DeepImpact, a new first-stage retrieval method that leverages a combination of a traditional inverted in dexes and contextualized language models for efficient retrieval. By estimating semantic importance, DeepImpact produces a single value impact score for each tokens of a document collection. Our results show that DeepImpact outperforms every inverted-index based baseline, in some cases even matching the effectiveness of more complex neural retrieval approaches such as ColBERT. Fur thermore, when ColBERT is used to re-rank candidates retrieved by DeepImpact

instead of approximate nearest neighbor, we find a dramatic reduction of query processing latency, and a more modest improvement in effectiveness of the whole pipeline. Future work will focus on further enhancing the underlying model. First, we would like to experiment with more relaxed matching conditions, instead of exact match, between the query-document terms.

[REF6] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Impact_score_learning/BIBREF91_4aa1d28944856ebe1950a27f633c6667ead3cbf8.pdf Title: Learning Passage Impacts for Inverted Indexes
Chunk of text: We conduct our experiments on the MS MARCO passage ranking dataset. To evaluate query processing effectiveness and efficiency, we compare with existing methods us ing the MSMARCO Dev Queries, 1 and we test all methods on the TREC 2019 and TREC 2020 queries from the TREC Deep Learning passage ranking track. Baselines. We perform two different sets of experiments.

[REF7] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Ranking_with_Encoder-decoder_Models/BIBREF41_9405cc0d6169988371b2755e573cc28650d14dfe.pdf Title: Language Models are Unsupervised Multask LearnersChunk of text: Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks have been proposed such as GLUE (Wang et al., 2018) and decaNLP (McCann et al., 2018) to begin studying this. Multitask learning (Caruana, 1997) is a promising frame work for improving general performance. However, mul titask training in NLP is still nascent. Recent work re ports modest performance improvements (Yogatama et al., 2019) and the two most ambitious efforts to date have trained on a total of 10 and 17 (dataset, objective) pairs respectively (McCann et al., 2018) (Bowman et al., 2018). From a meta-learning perspective, each (dataset, objective) pair is a single training example sampled from the distribution of datasets and objectives.

[REF8] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Impact_score_learning/BIBREF91_4aa1d28944856ebe1950a27f633c6667ead3cbf8.pdf Title: Learning Passage Impacts for Inverted Indexes
Chunk of text: DeepImpact is able to achieve sta tistically significant higher recall than all the compared methods (with one single exception at cutoff 1000). In particular, Table 4 shows that the gap with the other methods is greater with smaller cutoff values, which reduces the re-ranking cost and thus could enable the use of more complex pairwise ranking models, such as DuoBERT . In re-ranking, DeepImpact outperforms all other methods at cutoff 10. Moreover, it outperforms DeepCT on all cutoff values except 1000, and it is comparable with DocT5Query. Table 4: First-stage recall and re-rank-stage MRR@10 using ColBERT at several first-stage cutoffs for different candi date generation methods w.r.t. MSMARCO DevQueries. The symbol ∇ denotes a significant difference viz.

[REF9] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Vector_quantisation_approaches/BIBREF68_4748d22348e72e6e06c2476486a

fddbc76e5eca7.pdf Title: Product Quantization for Nearest Neighbor Search Chunk of text: First, the number of possible distances is significantly higher than for competing Hamming embedding methods , , , as the Hamming space used in these techniques allows for a few distinct distances only. Second, as a byproduct of the method, we get an estimation of the expected squared distance, which is required for ε-radius search or for using Lowe’s distance ratio criterion . The motivation of using the Hamming space in , , is to compute distances efficiently. Note, how ever, that one of the fastest ways to compute Hamming distances consists in using table lookups. Our method uses a similar number of table lookups, resulting in comparable efficiency. An exhaustive comparison of the query vector with all codes is prohibitive for very large datasets. We, therefore, introduce a modified inverted file structure to rapidly access the most relevant vectors.

.....
.....
.....
Title: Leveraging Contextual Neural Language Models for Deeper Text Understanding in Information Retrieval

Contextual neural language models have revolutionized various natural language processing tasks, including information retrieval. These models, such as BERT (Bidirectional Encoder Representations from Transformers), have shown remarkable performance in capturing the contextual information of text, leading to improved text understanding and retrieval [REF8]. In this section, we explore the potential of leveraging contextual neural language models for deeper text understanding in information retrieval.

One area where contextual neural language models have been successfully applied is in ranking effectiveness. Studies have shown that more sophisticated representation aggregation approaches, such as PARADE (Position-Aware Representation Aggregation for Document Retrieval), which utilizes transformer encoders, can significantly enhance ranking effectiveness [REF0]. PARADE and its variants, including PARADE CNN, have consistently demonstrated superior performance across different metrics, query types, and test collections [REF0]. These findings highlight the effectiveness of leveraging contextual neural language models for improved ranking in information retrieval.

Furthermore, the use of contextual neural language models extends beyond traditional information access tasks like ad hoc retrieval and question answering. It has been observed that the boundaries between text ranking, question answering, paraphrase detection, and other related problems are becoming increasingly blurred [REF1]. The connections between these tasks have become intertwined, with researchers adopting similar approaches and models to tackle them [REF1]. This convergence of natural language processing and information retrieval communities further emphasizes the significance of leveraging contextual neural language models for deeper text understanding in information retrieval.

In addition to ranking effectiveness, contextual neural language models offer potential benefits in terms of efficiency. Knowledge

distillation, a technique that involves transferring knowledge from a large teacher model to a smaller student model, has been successfully applied to text ranking tasks [REF8]. This approach allows for effective control of the tradeoff between effectiveness and efficiency, making it particularly valuable in resource-constrained scenarios [REF8]. By distilling the knowledge from a large pretrained teacher model, smaller student models can achieve comparable performance while reducing computational requirements [REF8].

The application of contextual neural language models in information retrieval also raises interesting questions about the importance of position information. While position information has been considered important in information retrieval, studies have shown that its impact may be limited, especially in small collections and high recall situations [REF5]. The gains observed from utilizing position information are often small, suggesting that other factors, such as semantic similarity and relevance, play a more significant role in text understanding and retrieval [REF5].

In conclusion, leveraging contextual neural language models holds great promise for deeper text understanding in information retrieval. These models have demonstrated their effectiveness in improving ranking performance and have the potential to enhance efficiency through techniques like knowledge distillation. However, further research is needed to explore the full potential of these models and to better understand the role of position information in information retrieval tasks.

References given to GPT:
[REF0] - paperID: ./papers_pdf/paper_section/Conclusions/BIBREF6_2c953a3c378b40dadf2e3fb486713c8608b8e282.pdf Title: Pretrained Transformers for Text Ranking: BERT and Beyond Chunk of text: We see that, in general, ranking effectiveness increases with more sophisticated representation aggregation approaches. The experimental results suggest the following conclusions: • PARADE (6f), which performs aggregation using transformer encoders, and PARADE CNN (6e) are consistently the most effective across different metrics, query types, and test collections. PARADECNN usually performs slightly worse than the full PARADE model, but the differences are not statistically significant. • PARADE Avg (6a) is usually the least effective. • PARADESum (6b) and PARADE Attn (6d) perform similarly; PARADESum is slightly more effective on Robust04 and PARADE Attn is slightly more effective on Gov2. PARADESum can be viewed as PARADE Attn with uniform attention weights, so this result suggests that the attention scores produced by PARADE Attn may not be necessary.

[REF1] - paperID: ./papers_pdf/paper_section/Conclusions/BIBREF6_2c953a3c378b40dadf2e3fb486713c8608b8e282.pdf Title: Pretrained Transformers for Text Ranking: BERT and Beyond Chunk of text: We can broadly characterize ad hoc retrieval, question answering, and the different tasks described above as “information access”—a term we use to refer to these technologies collectively. Text ranking is without a doubt an important component of information access. However, beyond information access, examples of text ranking abound

in natural language processing. For example: Semantic Similarity Comparisons. The question of whether two texts “mean the same thing” is a fundamental problem in natural language processing and closely related to the question of whether a text is relevant to a query. While there are some obvious differences, researchers have explored similar approaches and have often even adopted the same models to tackle both problems. In the context of learned dense representations for ranking, the connections between these two problems have become even more intertwined, bringing the NLP and IR communities closer and further erasing the boundaries between text ranking, question answering, paraphrase detection, and many related problems.

[REF2] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Optimisations/BIBREF73_2cbb8de53759e75411bc528518947a3094fbce3a.pdf
Title: Billion-Scale Similarity Search with GPUs Chunk of text: 4.2 WarpSelect Our k-selection implementation, WarpSelect, maintains state entirely in registers, requires only a single pass over data and avoids cross-warp synchronization. It uses merge odd and sort-odd as primitives. Since the register file pro vides much more storage than shared memory, it supports k ≤ 1024. Each warp is dedicated to k-selection to a single one of the n arrays [ai]. If n is large enough, a single warp per each [ai] will result in full GPU occupancy. Large ` per warp is handled by recursive decomposition, if ` is known in advance.

[REF3] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-BOW_Encodings/BIBREF10_47ced790a563344efae66588b5fb7fe6cca29ed3.pdf
Title: The Probabilistic Relevance Framework: BM25 and Beyond
Chunk of text: This worked well in practice, but it can lead to degener ate cases (e.g., when a stream is extremely verbose and contains most terms for most documents). The proper definition of IDF in this con text requires further research (this is also discussed in  where the notion of an expected idf is introduced).����3.7 Non-Textual Relevance Features 365 Table 3.1. BM25F parameters reported in  for topic distillation (TD) and name page (NP) search tasks Parameter TD’03 NP’03 k1 27.5 4.9 btitle 0.95 0.6 bbody 0.7 0.5 banchor 0.6 0.6 vtitle 38.4 13.5 vbody 1.0 1.0 vanchor 35 11.5 As an illustration, we report in Table 3.1 the BM25Fweights reported in  for the 2003 TREC Web Search tasks. 3.6.4 Interpretation of the Simple Version It is worth mentioning a very transparent interpretation of the simple version — although it does not apply directly to the version with vari able b, it may give some insight. If the stream weights vs are inte gers, then we can see the simple BM25F formula as an ordinary BM25 function applied to a document in which some of the streams have been replicated. For example, if the streams and weights are {vtitle = 5, vabstract = 2, vbody = 1}, then formula 3.18 is equivalent to 3.15 applied to a document in which the title has been replicated five times and the abstract twice. 3.7 Non-Textual Relevance Features In many collections there are other sources of relevance information besides the text.

[REF4] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-

BOW_Encodings/BIBREF10_47ced790a563344efae66588b5fb7fe6cca29ed3.pdf
Title: The Probabilistic Relevance Framework: BM25 and Beyond
Chunk of text: Then, given eliteness (from the author's choice of topics to cover), the unigram probabilities for the language model for filling the term-positions would also be stream-specific. In particular, there would be a much stronger bias to the elite terms when choosing words for the title than for the body (we expect a title to be much denser in topic-specific terms than an average body sentence). The consequence of this term-document eliteness property is that we should combine evidence across terms and streams in the opposite order to that suggested above: first streams, then terms. That is, for each term, we should accumulate evidence for eliteness across all the streams. The saturation function should be applied at this stage, to the total evidence for each term. Then the final document score should be derived by combination across the terms. 3.6.2 Notation We have a set of S streams, and we wish to assign relative weights vs to them.

[REF5] - paperID:
./papers_pdf/paper_section/Text_Representations_for_Ranking-BOW_Encodings/BIBREF10_47ced790a563344efae66588b5fb7fe6cca29ed3.pdf
Title: The Probabilistic Relevance Framework: BM25 and Beyond
Chunk of text: We end this section with a brief discussion of why position infor mation may not be as important as it may seem at first view. It is sobering to see how hard it has been in the past to effectively use prox imity in IR experiments. All the works referenced in this section claim statistically significant improvements over non-positional baselines, but the improvements reported are small. We believe this is specially the case for small collections and high recall situations (typical of academic IR evaluations), since position information is a precision enhancement technique. But even in large collections with high-precision require ments (such as realistic Web Search evaluations) the gains observed are small. Why is this? We do not know of theoretical or empirical studies about this, but we propose here two hypotheses.

[REF6] - paperID:
./papers_pdf/paper_section/Text_Representations_for_Ranking-BOW_Encodings/BIBREF5_3cf0822f63e51be5343028bad7ee72a5882ef7de.pdf
Title: Scalability Challenges in Web Search Engines Chunk of text: [51, 113]. The crawler aims to locate and fetch as many pages as possible from the Web. This way, it increases the likelihood that more pages useful to users will be indexed by the search engine. In the mean time, the crawler tries to keep the pages that are already discovered as fresh as possible by selectively refetching them, as an effort towards providing pages' up to-date versions in the Web, rather than their stale versions in the repository. Finally, the crawler tries to prioritize fetching of pages in such a way that relatively more im portant pages are downloaded earlier or are fetched more often, keeping them more fresh compared to less important pages . Achieving the above-mentioned quality objectives re quires sustaining high page download rates, which is the most important efficiency objective for the crawler. This is simply because, as the crawler downloads pages faster, it can cope better with the growth and evolution of the Web.

[REF7] - paperID: ./papers_pdf/paper_section/Conclusions/BIBREF97_b97a33933541c276778c3fe63baad6964f4bdf44.pdf Title: Neural Approaches to Conversional Information Retrieval Chunk of text: The section of Table 2.2 about finding answers lists several other kinds of dataset that are worth noting, several of which are also described in other parts of the book. The content given in some cases is a paragraph of text, which means we assume some process has already taken place to find a paragraph of interest to the user. In other cases, the content is structured data, such as Wikipedia tables, Wikidata knowledge triples and databases. We also see cases where the content is a text corpus, so we are either searching a corpus of full documents or a corpus of shorter passages of text. In all cases the annotated output is shorter than a full document, reflecting the goal of a conversational system to identify outputs that are short enough for presentation to the user on a small screen or through voice. The answer can be a passage of text such as a sentence, or it can even be a span of text that gives a direct answer. Some datasets identify both the passage and the direct answer.

[REF8] - paperID: ./papers_pdf/paper_section/Conclusions/BIBREF6_2c953a3c378b40dadf2e3fb486713c8608b8e282.pdf Title: Pretrained Transformers for Text Ranking: BERT and Beyond Chunk of text: Knowledge distillation is a general-purpose approach to controlling effective ness/efficiency tradeoffs with neural networks. It has previously been demonstrated for a range of natural language processing tasks, and recent studies have applied the approach to text ranking as well. While knowledge distillation inevitably degrades effectiveness, the potentially large increases in efficiency make the tradeoffs worthwhile under certain operating scenarios. Emerging evidence suggests that the best practice is to distill a large teacher model that has already been fine-tuned for ranking into a smaller pretrained student model. 3.5.2 Ranking with Transformers: TK, TKL, CK Empirically, BERT has proven to be very effective for many NLP and information access tasks. Combining this robust finding with the observation that BERT appears to be over-parameterized (for example, Kovaleva et al. ) leads to the interesting question of whether smaller models might be just as effective, particularly if limited to a specific task such as text ranking.

[REF9] - paperID: ./papers_pdf/paper_section/Conclusions/BIBREF97_b97a33933541c276778c3fe63baad6964f4bdf44.pdf Title: Neural Approaches to Conversional Information Retrieval Chunk of text: report good performance on the TREC CAsT 2019 dataset, where the majority of expansion terms in the oracle manual de-contextualized queries (see the example in Table 3.2) are indeed from the previous query turns. However, it remains to be validated whether the observation and good performance hold on other conversational search benchmarks or in real-world scenarios. 3.5.3 Neural Query Rewriting The query rewriting approach to CQU uses a neural natural language generation (NLG) model to generate standalone query $Q\hat{}\ i$ , using the conversational query $Qi$ and its dialog history H$i$ . In the TREC CAsT setting, we have H$i$ = {$Q$1, ..., $Qi$−1}. Thus, the approach can be written as {$Q$1, ..., $Qi$−1}, $Qi$ NLG( $\theta$) ------→ $Q\hat{}\ i$ , (3.12)

45Figure 3.8: A conversational search session (Left) and an ad hoc search session (Right), adapted from Yu et al.  .

.....
.....
.....
Title: Advancements in Dense Retrieval for Information Retrieval

Dense retrieval has emerged as a promising approach in information retrieval, leveraging dense vector representations to improve the effectiveness of retrieval systems. This section explores recent advancements in dense retrieval for information retrieval, focusing on key techniques and methodologies that have contributed to its success.

One important aspect of dense retrieval is the construction of the K-Nearest Neighbor Graph (K-NNG) [REF0]. The K-NNG is a directed graph that connects each object in a set to its K most similar objects under a given similarity measure. The construction of the K-NNG is crucial for various web-related applications, such as collaborative filtering and content-based search systems. The K-NNG enables efficient recommendations and offline search operations, making it a key data structure in data mining and machine learning [REF0].

To ensure the effectiveness of dense retrieval models, the informativeness of constructed negatives plays a vital role [REF1]. In the training process, negative samples with near-zero loss and gradients contribute little to model convergence. Therefore, it is essential to select informative negatives that are hard to distinguish from positive samples. However, relying solely on in-batch local negatives may not provide sufficiently informative samples due to the properties of text retrieval. The batch size is typically smaller than the corpus size, and only a few negatives are informative, while the majority of the corpus is unrelated [REF1].

Query term weighting is another area where advancements have been made in dense retrieval. DeepTR [REF2] proposes using estimated query term weights to generate bag-of-words queries (BOW) and sequential dependency model queries (SDM). By re-formulating the original BOW query with predicted weights, the query's effectiveness can be enhanced. Additionally, the sequential dependency model incorporates bigrams and word co-occurrences within a window to improve query performance. Terms with non-positive weights are discarded, ensuring the relevance of the generated queries [REF2].

Efficient construction of the K-NNG is crucial for dense retrieval systems. The NN-Descent algorithm [REF7] provides a scalable and space-efficient approach for constructing the K-NNG. By incorporating optimizations such as neighborhood pruning, early termination, and candidate set reduction, NN-Descent achieves high recall while maintaining reasonable computational costs. The algorithm is independent of similarity measures, making it applicable to various domains and retrieval tasks [REF7].

Evaluation of dense retrieval models requires appropriate test datasets. The TREC-CAR dataset [REF5] has been widely used for evaluating dense retrieval models. It consists of a large collection of English Wikipedia passages, with synthetic queries and relevant passages generated for evaluation purposes. The dataset provides a diverse range of queries and passages, enabling comprehensive assessment of dense retrieval models' performance [REF5].

DeepCT-Query [REF6] introduces a query term weighting framework that leverages deep neural networks to estimate term importance based on the context's meaning. By training DeepCT with relevant query-document pairs, weighted queries can be generated and used with popular retrieval models such as BM25 and query likelihood. DeepCT's ability to differentiate between central and non-central terms leads to significant improvements in retrieval accuracy, particularly for long queries that mention multiple terms and concepts [REF6].

In conclusion, advancements in dense retrieval for information retrieval have significantly improved the effectiveness and efficiency of retrieval systems. Techniques such as K-NNG construction, informative negative sampling, query term weighting, and efficient algorithms for graph construction have contributed to the success of dense retrieval models. These advancements have paved the way for more accurate and scalable information retrieval systems in various domains.

References given to GPT:
[REF0] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Graph_approaches/BIBREF69_f17c6e164ccc7ec1ad91b3fbbafe8f84664e9803.pdf  Title: Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures Chunk of text: INTRODUCTION The K-Nearest Neighbor Graph (K-NNG) for a set of ob jects V is a directed graph with vertex set V and an edge from each v ∈ V to its K most similar objects in V under a given similarity measure, e.g. cosine similarity for text, l2 distance of color histograms for images, etc. K-NNG con struction is an important operation with many web related applications: in (user-based) collaborative filtering , a K NNG is constructed by connecting users with similar rating patterns, and used to make recommendations based on the active user’s graph neighbors; in content-based search sys tems, when the dataset is fixed, a K-NNG constructed of fline is more desirable than the costly online K-NN search. K-NNG is also a key data structure for many established Copyright is held by the International World Wide Web Conference Com mittee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others. WWW 2011, March 28–April 1, 2011, Hyderabad, India. ACM 978-1-4503-0632-4/11/03. methods in data mining  and machine learning , es pecially manifold learning .

[REF1] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Optimisations/BIBREF51_c9b8593db099869fe7254aa1fa53f3c9073b0176.pdf

Title: Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval Chunk of text: l(d +, d−) → 0 ⇒ ||∇φL l(d +, d−)||2 → 0 ⇒ ||∇θt l(d +, d−)||2 → 0. (12) Intuitively, negative samples with near zero loss have near zero gradients and contribute little to model convergence. The convergence of dense retrieval model training relies on the informativeness of constructed negatives. Inefficacy of Local In-Batch Negatives: We argue that the in-batch local negatives are unlikely to provide informative samples due to two common properties of text retrieval. Let D−∗ be the set of informative negatives that are hard to distinguish from D+, and b be the batch size, we have (1) b  |C|, the batch size is far smaller than the corpus size; (2) |D−∗|  |C|, that only a few negatives are informative and the majority of corpus is trivially unrelated.

[REF2] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Impact_score_learning/BIBREF89_3de1752cd0854e220fc41f0ccf7db913f846284c.pdf  Title: Context-Aware Sentence/Passage Term Importance Estimation for First Stage Retrieval  Chunk of text: Following DeepTR , we use the estimated query term weights to generate bag-of-words queries (BOW) and sequential dependency model queries (SDM). For example, in the Indri  query language, the original BOW query "apple pie" is re-formulated into #weight(0.8 apple 0.7 pie)3 for predicted weights apple:0.8, pie:0.7. Sequential dependency model adds bigrams and word co-occurrences within a window to the query. We use the re-weighted BOW query to replace the bag-of words part of the SDM query. Terms with non-positive weights are 3 #weight is Indri’s probabilistic weighted-AND operator. discarded.

[REF3] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Graph_approaches/BIBREF69_f17c6e164ccc7ec1ad91b3fbbafe8f84664e9803.pdf  Title: Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures Chunk of text: Any v ′ ∈ Br/2(v) is likely to satisfy this requirement, as we have: 1. v ′ is also in Br(v), so Pr {v ′ ∈ B[v]} ≥ K/|Br(v)|. 2. d(u, v′ ) ≤ d(u, v) + d(v, v′ ) ≤ r, so Pr {u ∈ B[v ′ ]} ≥ K/|Br(v ′ )|. 3. |Br(v)| ≤ c|Br/2(v)|, and |Br(v ′ )| ≤ c|Br/2(v ′ ) | ≤ c|Br(v)| ≤ c 2 |Br/2(v)|. Combining 1—3 and assuming independence, we get Pr {v ′ ∈ B[v] ∧ u ∈ B[v ′ ]} ≥ K/|Br/2(v)| 2 In total, we have |Br/2(v)| candidates for such v ′ , so that Pr {u ∈ B ′

[REF4] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Graph_approaches/BIBREF70_c197ecb6a6987667cadcb498136989af1827cce0.pdf  Title: Approximate Nearest Neighbor Algorithm based on Navigable Small World GraphsChunk of text: The tests indicate  that at least for Euclid data with d¼1...20, the optimal value for number of neighbors to connect (f) is about 3d, making memory consumption linear with the dimensionality. Lesser values of f can be used to reduce the complexity of a single search, sacrifi cing its recall quality. 6. Test results and discussion 6.1. Test data We have implemented the algorithms presented above in order to validate our assumptions about the scalability of the structure,

and to evaluate its performance. In our tests we have used a workstation based on two Intel Xeon X5675 six core processors with 192 Gb of RAM. The algorithm was written in Java using the Oracle Java Platform. We have used the following test datasets: Uniformly distributed random points with L2 (Eucli dean distance) distance function (up to 5  107 ele ments, up to 50 dimensions).

[REF5] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Impact_score_learning/BIBREF89_3de1752cd0854e220fc41f0ccf7db913f846284c.pdf  Title: Context-Aware Sentence/Passage Term Importance Estimation for First Stage Retrieval  Chunk of text: TREC-CAR consists of 29.7M English Wikipedia passages with an average length of 61 words. Queries and relevant pas sages are generated synthetically. A query is the concatenation of a Wikipedia article title with the title of one of its sections. Following prior work [22, 23], we use the automatic relevance judg ments, which treats paragraphs within the section as relevant to the query. The training set and validation set have 3.3M query-passage pairs and 0.8M pairs respectively. The test query set contains 1,860 queries with an average of 2.5 relevant paragraphs per query.

[REF6] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Impact_score_learning/BIBREF89_3de1752cd0854e220fc41f0ccf7db913f846284c.pdf  Title: Context-Aware Sentence/Passage Term Importance Estimation for First Stage Retrieval  Chunk of text: For long queries that mention many terms and concepts, it is important to identify which are central. We follow a query term weighting framework proposed by Zheng and Callan  to develop DeepCT-Query. It trains DeepCT with signals from relevant query-document pairs, weighting query terms by their possibilities to be mentioned by relevant documents. The predic tions are used to generate weighted queries that can be used with widely-used retrieval models such as BM25  and query likeli hood . Our experiments demonstrate that DeepCT generates effective representations for both passages and queries that lead to large improvements in first-stage retrieval accuracy. Analysis shows that DeepCT’s main advantage is its ability to estimate term importance using the meaning of the context rather than term frequency signals, allowing the retrieval model to differentiate between key terms and other frequently mentioned but non-central terms.

[REF7] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Graph_approaches/BIBREF69_f17c6e164ccc7ec1ad91b3fbbafe8f84664e9803.pdf  Title: Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures Chunk of text: The Full Algorithm The full NN-Descent algorithm incorporating the four op timizations discussed above is listed in Algorithm 2. In this paper we are mainly interested in a method that is inde pendent of similarity measures. Optimizations specialized to particular similarity measures are possible. For exam ple, if the similarity measure is a distance metric, triangle inequality could be potentially used to avoid unnecessary computation. Our optimizations are not sufficient to ensure that the similarity between two objects is only evaluated once. Full elimination of redundant computation would require a

table of O(N 2 ) space, which is too expensive for large datasets. Space efficient approximations, like Bloom filter, are pos sible, but come with extra computational cost, and would only be helpful if similarity measure is very expensive to compute.

[REF8] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Graph_approaches/BIBREF69_f17c6e164ccc7ec1ad91b3fbbafe8f84664e9803.pdf  Title: Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures Chunk of text: Our method works with an arbitrary simi-larity oracle — a function that produces a similarity score for two objects. • Scalable. As the size of the dataset grows, our method only sees a marginal decline in recall, and the empir ical cost is around O(n 1.14) for all datasets we exper imented with. Our method mainly operates on infor mation that is local to each data item, and is intrinsi cally suitable for a distributed computing environment like MapReduce . • Space efficient. In principle, the only data structure we need is an approximate K-NNG which is also the final output: our method can iteratively improve the graph in place. For optimization, or in a distributed implementation, minimal extra data are maintained.

[REF9] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Impact_score_learning/BIBREF89_3de1752cd0854e220fc41f0ccf7db913f846284c.pdf  Title: Context-Aware Sentence/Passage Term Importance Estimation for First Stage Retrieval  Chunk of text: DeepCT successfully transfers the text understanding ability from a deep neural network into simple signals that can be efficiently consumed by early-stage ranking systems and boost their performance. Analysis shows the main advantage of DeepCT over classic term weighting approaches: DeepCT finds the most central words in a text even if they are mentioned only once. Non-central words, even if mentioned frequently in the text, are suppressed. Such behavior is uncommon in previous term weighting approaches. We view DeepCT as an encouraging step from “frequencies” to “ meanings”.

.....
.....
.....
Title: Learning to Rank: A Comprehensive Overview for Information Retrieval

Learning to rank (LTR) is a prominent approach in information retrieval (IR) that aims to produce a ranking of relevant documents given a query and a document collection. LTR models evaluate the interactions between queries and documents, assigning higher scores to documents that better match the query [REF0]. Traditional LTR models rely on handcrafted features to encode query-document interactions, such as relevance scores from unsupervised ranking models. However, recent advancements in deep neural models have introduced the possibility of directly extracting interactions based on queries and documents [REF0].

Early neural IR models can be categorized as semantic matching models, where both queries and documents are embedded into a low-

dimensional space, and their similarity is assessed based on dense representations [REF0]. Examples of such models include DSSM (Deep Structured Semantic Model) and DESM (Deep Embedding Semantic Model) [REF0]. These models have shown the capacity to learn linguistic patterns and capture semantic relationships between words [REF2].

To evaluate the performance of LTR models, various evaluation metrics have been proposed. One approach is to convert graded judgments into ranked document pairs and compare the predictions of the models with the ground truth labels [REF1]. Another approach is to measure the accuracy of the models in predicting relevance levels for different pairs of labels [REF1]. These evaluation methods allow for a comprehensive comparison of different LTR models and provide insights into their strengths and weaknesses.

While LTR models have shown promising results, there are still open questions regarding the extent to which deep neural models can improve IR and the components that should be included in a deep neural model for IR [REF0]. The choice of neural network structure, the selection of appropriate distance measures, and the consideration of hyperparameters are among the factors that need to be carefully addressed [REF4] [REF7]. Additionally, the influence of different types of data, such as linguistic corpora and non-linguistic features, on the performance of LTR models is an area of ongoing research [REF3].

In recent years, models like GloVe (Global Vectors for Word Representation) have been proposed to learn word representations that capture both count data and meaningful linear substructures [REF5]. These models have demonstrated superior performance on word analogy, word similarity, and named entity recognition tasks [REF5]. Incorporating such models into LTR frameworks could potentially enhance the effectiveness of LTR models in capturing semantic relationships between queries and documents.

In conclusion, LTR is a powerful approach in IR that aims to produce rankings of relevant documents based on query-document interactions. Recent advancements in deep neural models have opened up new possibilities for directly extracting interactions from queries and documents. However, there are still challenges to be addressed, such as the selection of appropriate network structures, distance measures, and hyperparameters. Incorporating models that capture both count data and meaningful substructures could further improve the performance of LTR models in capturing semantic relationships.

References given to GPT:
[REF0] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Convolutional_Neural_Networks/BIBREF28_32e7f0863e7c56cfced89abedaee46e2288bc127.pdf Title: PACRR: A Position-Aware Neural IR Model for Relevance Matching Chunk of text: Introduction Despite the widespread use of deep neural mod els across a range of linguistic tasks, to what extent such models can improve information re trieval (IR) and which components a deep neural model for IR should include remain open ques tions. In ad-hoc IR, the goal is to produce a rank ing of relevant documents given an open-domain (“ad

hoc”) query and a document collection. A ranking model thus aims at evaluating the inter actions between different documents and a query, assigning higher scores to documents that better match the query. Learning to rank models, like the recent IRGAN model (Wang et al., 2017), rely on handcrafted features to encode query docu ment interactions, e.g., the relevance scores from unsupervised ranking models. Neural IR mod els differ in that they extract interactions directly based on the queries and documents. Many early neural IR models can be categorized as seman tic matching models, as they embed both queries and documents into a low-dimensional space, and then assess their similarity based on such dense representations. Examples in this regard include DSSM (Huang et al., 2013) and DESM (Mitra et al., 2016).

[REF1] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Convolutional_Neural_Networks/BIBREF28_32e7f0863e7c56cfced89abedaee46e2288bc127.pdf Title: PACRR: A Position-Aware Neural IR Model for Relevance Matching Chunk of text: Thus, it is possible for us to compare different models over the same set of complete judgments, remov ing the issue of different initial runs. Moreover, although ranking is our ultimate target, a direct in spection of pairwise prediction results can indicate which kinds of document pairs a model succeeds at or fails on. We first convert the graded judg ments from TREC into ranked document pairs by comparing their labels. Document pairs are cre ated among documents that have different labels. A prediction is counted as correct if it assigns a higher score to the document from the pair that is labeled with a higher degree of relevance. The judgments from TREC contain at most six rele vance levels, and we merge and unify the original levels from the six years into four grades, namely, Nav, HRel, Rel and NRel. We compute the ac curacy for each pair of labels.

[REF2] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-Word_Embeddings/BIBREF20_f37e1b62a767a307c046404ca96bc140b3e68cb5.pdf Title: GloVe: Global Vectors for Word Representation Chunk of text: Recently, the importance of the full neural net work structure for learning useful word repre sentations has been called into question. The skip-gram and continuous bag-of-words (CBOW) models of Mikolov et al. (2013a) propose a sim ple single-layer architecture based on the inner product between two word vectors. Mnih and Kavukcuoglu (2013) also proposed closely-related vector log-bilinear models, vLBL and ivLBL, and Levy et al. (2014) proposed explicit word embed dings based on a PPMI metric. In the skip-gram and ivLBL models, the objec tive is to predict a word’s context given the word itself, whereas the objective in the CBOW and vLBL models is to predict a word given its con text. Through evaluation on a word analogy task, these models demonstrated the capacity to learn linguistic patterns as linear relationships between the word vectors. Unlike the matrix factorization methods, the shallow window-based methods suffer from the disadvantage that they do not operate directly on the co-occurrence statistics of the corpus.

[REF3] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-Word_Embeddings/BIBREF17_5303f288c0de1fc717c3389773a2a684589ee46b.p

df  Title: Semantic Memory Search and Retrieval in a Novel Cooperative Word Game: A Comparison of Associative and Distributional Semantic Models  Chunk of text: The present results highlight how associative models do indeed emphasize semantic rela tionships not well-represented within the DSMs and indicate that reliance on pure linguistic corpora within the DSMs may not be sufficient to capture the variety of responses produced by participants in the Connector game. Indeed, in addition to the linguistic content of free associations, associative responses also tend to reflect experiences that evoke mental imagery and emotional responses (De Deyne et al., 2021). It is possible that similar representations are activated when speakers and guessers are searching through semantic space within the Con nector game, which the associative models tend to capture. DSMs have been criticized for relying solely on linguistic corpora and therefore their inability to capture non-linguistic fea tures of meaning (Barsalou, 2016; De Deyne et al., 2016). Our results also shed light on some additional aspects of meaning (e.g., hierarchical relationships) that may be readily appar ent to humans (and are therefore well-represented in the associative models) but are missing from the DSMs. Within this context, associative models may provide an important behavioral baseline or benchmark for comparisons across DSMs and may therefore be useful in assess ing the psychological plausibility of different DSMs (for a detailed discussion, see Kumar, 2021). Indeed, the present work highlights systematic differences across two popular DSMs (GloVe and word2vec) in accounting for performance in the game, with GloVe outperforming word2vec in the speaker task.

[REF4] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-Word_Embeddings/BIBREF20_f37e1b62a767a307c046404ca96bc140b3e68cb5.pdf  Title: GloVe: Global Vectors for Word Representation  Chunk of text: One could inter pret this objective as a “global skip-gram” model, and it might be interesting to investigate further. On the other hand, Eqn. (13) exhibits a number of undesirable properties that ought to be addressed before adopting it as a model for learning word vectors. To begin, cross entropy error is just one among many possible distance measures between prob ability distributions, and it has the unfortunate property that distributions with long tails are of ten modeled poorly with too much weight given to the unlikely events. Furthermore, for the mea sure to be bounded it requires that the model dis tribution Q be properly normalized. This presents a computational bottleneck owing to the sum over the whole vocabulary in Eqn. (10), and it would be desirable to consider a different distance measure that did not require this property of Q. A natural choice would be a least squares objective in which normalization factors in Q and P are discarded, Jˆ = X i, j Xi  Pˆ i

[REF5] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-Word_Embeddings/BIBREF20_f37e1b62a767a307c046404ca96bc140b3e68cb5.pdf  Title: GloVe: Global Vectors for Word Representation  Chunk of text: We construct a model that utilizes this main ben efit of count data while simultaneously capturing the meaningful linear substructures prevalent in recent log-bilinear prediction-based methods like word2vec. The result, GloVe, is a new global log-

bilinear regression model for the unsupervised learning of word representations that outperforms other models on word analogy, word similarity, and named entity recognition tasks. Acknowledgments We thank the anonymous reviewers for their valu able comments. Stanford University gratefully acknowledges the support of the Defense Threat Reduction Agency (DTRA) under Air Force Re search Laboratory (AFRL) contract no. FA8650- 10-C-7020 and the Defense Advanced Research Projects Agency (DARPA) Deep Exploration and Filtering of Text (DEFT) Program under AFRL contract no. FA8750-13-2-0040.

[REF6] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-Word_Embeddings/BIBREF20_f37e1b62a767a307c046404ca96bc140b3e68cb5.pdf Title: GloVe: Global Vectors for Word Representation Chunk of text: Named entity recognition. The CoNLL-2003 English benchmark dataset for NER is a collec tion of documents from Reuters newswire articles, annotated with four entity types: person, location, organization, and miscellaneous. We train mod els on CoNLL-03 training data on test on three datasets: 1) ConLL-03 testing data, 2) ACE Phase 2 (2001-02) and ACE-2003 data, and 3) MUC7 Formal Run test set. We adopt the BIO2 annota tion standard, as well as all the preprocessing steps described in (Wang and Manning, 2013). We use a comprehensive set of discrete features that comes with the standard distribution of the Stanford NER model (Finkel et al., 2005). A total of 437,905 discrete features were generated for the CoNLL 2003 training dataset. In addition, 50-dimensional vectors for each word of a five-word context are added and used as continuous features.

[REF7] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Convolutional_Neural_Networks/BIBREF28_32e7f0863e7c56cfced89abedaee46e2288bc127.pdf Title: PACRR: A Position-Aware Neural IR Model for Relevance Matching Chunk of text: However, to gain a better understanding of the influence of different hyper-parameters, we ex plore PACRR-kwindow's effectiveness when sev eral hyper-parameters are varied. The results when re-ranking QL search results are given in Figure 3. The results are reported based on the models with the highest validation scores after fixing certain hyper-parameters. For example, the ERR@20 in the leftmost figure is obtained when fixing ld to the values shown. The crosses in Figure 3 corre spond to the models that were selected for use on the test data, based on their validation set scores. It can be seen that the selected models are not neces sarily the best model on the test data, as evidenced by the differences between validation and test data results, but we consistently obtain scores within a reasonable margin. Owing to space constraints, we omit the plots for PACRR-firstk.

[REF8] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Convolutional_Neural_Networks/BIBREF28_32e7f0863e7c56cfced89abedaee46e2288bc127.pdf Title: PACRR: A Position-Aware Neural IR Model for Relevance Matching Chunk of text: In particular, we present a novel re-ranking model called PACRR (Position-Aware Convolutional Recurrent Relevance Matching). Our approach first produces similarity matrices that record the semantic similarity

between each query term and each individual term occurring in a document. These matrices are then fed through a series of convolutional, max-k-pooling, and recurrent lay ers so as to capture interactions corresponding to, for instance, bigram and trigram matches, and fi nally to aggregate the signals in order to produce global relevance assessments. In our model, the convolutional layers are designed to capture both unigram matching and positional information over text windows with different lengths; k-max pool ing layers are along the query dimension, preserv ing matching signals over different query terms; the recurrent layer combines signals from differ ent query terms to produce a query-document rel evance score. Organization. The rest of this paper unfolds as follows. Section 2 describes our approach for computing similarity matrices and the architecture of our deep learning model.

[REF9] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Convolutional_Neural_Networks/BIBREF24_563e821bb5ea825efb56b77484f5287f08cf3753.pdf Title: Convolutional Networks for Images, Speech, and Time-Series  Chunk of text: But, the main de ciency of unstructured nets for image or speech aplications is that they have no built-in invariance with respect to translations, orLeCun & Bengio: Convolutional Networks for Images, Speech, and Time-Series 4 local distortions of the inputs. Before being sent to the  xed-size input layer of a neural net, character images, spoken word spectra, or other 2D or 1D signals, must be approximately size-normalized and centered in the input  eld. Unfortunately, no such preprocessing can be perfect: handwriting is often normalized at the word level, which can cause size, slant, and position variations for individual characters; words can be spoken at varying speed, pitch, and intonation. This will cause variations in the position of distinctive features in input ob jects. In principle, a fully-connected network of su cient size could learn to produce outputs that are invariant with respect to such variations. However, learning such a task would probably result in multiple units with identical weight patterns positioned at various locations in the input. Learning these weight con gurations requires a very large number of training instances to cover the space of possible variations.

.....
.....
.....
Title: A Deep Look into Neural Ranking Models for Information Retrieval

Neural ranking models have gained significant attention and popularity in the field of information retrieval in recent years [REF2]. These models leverage deep learning techniques to improve the effectiveness and efficiency of ranking documents in response to user queries. In this section, we will delve into the various aspects of neural ranking models, including their practical effectiveness, training paradigms, and novel applications.

One of the key challenges in information retrieval is the ranking of search results based on user queries [REF3]. Traditional retrieval models, such as Okapi BM25 and Statistical Language Models, have been widely used as the backbone for search ranking [REF3]. However, with the advent of neural ranking models, researchers have explored new approaches that go beyond the traditional models and leverage the power of deep learning.

Compared to traditional retrieval models, neural ranking models offer several advantages. They have the potential to outperform state-of-the-art models that rely on hand-crafted features [REF2]. These models have been successfully applied to various ranking tasks, including ad-hoc retrieval, community-based QA, and conversational search [REF2]. Furthermore, researchers have started to explore new training paradigms and indexing schemes for neural ranking models, as well as the integration of external knowledge [REF2]. These advancements have led to exciting progress in the field of neural ranking models.

Evaluation of neural ranking models is crucial to assess their effectiveness. Researchers have used various metrics to evaluate the performance of these models. For instance, perplexity is commonly used to measure the fluency of language models, reflecting how well the model can generate the correct next word given the preceding words [REF1]. Prompt ranking accuracy is another metric used to assess how strongly a model's output depends on its input [REF1]. By measuring the likelihood of generated stories under different prompts, researchers can determine the percentage of cases where the true prompt is the most likely to generate the story [REF1].

While neural ranking models have shown promising results, it is important to note that their performance can be unstable across different collections [REF0]. In some cases, the performance may even be worse than that of traditional language models [REF0]. Therefore, it is crucial to carefully evaluate the practical effectiveness of neural ranking models on different ranking tasks and collections [REF2].

In conclusion, neural ranking models have emerged as a powerful approach for information retrieval, offering improved performance compared to traditional retrieval models [REF2]. These models have been applied to various ranking tasks and have shown exciting progress in terms of effectiveness and efficiency [REF2]. However, their performance can be unstable across different collections, highlighting the need for further evaluation and research [REF0].

References given to GPT:
[REF0] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Document_expansion_learning/BIBREF79_1225eb6570ce8d45067329fafcc8ff7636a65923.pdf  Title: Modeling and Solving Term Mismatch for Full-Text Retrieval  Chunk of text: Compared to the new features obtaining stable and significant improvement over the Language Model baseline, predictions based only on idf results in an unstable gain. Across most collections, performance improvement is not significant, and in one collection performance is worse than

that of the baseline lan guage model. Only 3 out of the 8 cases show statistically significant improvement. Clarity is traditionally used to predict query performance, thus we include another baseline that pre dicts recall based on term clarity as the only feature (row “Clarity” of Table 6.9). Performance is also unstable across collections, and on average is similar to using only idf to predict. 6.2.4.6 Comparing to the Relevance Model The Relevance Model theory (Lavrenko and Croft 2001) suggests to use term relevance probabilities as user term weights, which we follow, as shown in Section 6.1.2.

[REF1] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Document_expansion_learning/BIBREF85_29de7c0fb3c09eaf55b20619bceaeafe72fd87a6.pdf  Title: Hierarchical Neural Story Generation  Chunk of text: We do not aim to generate a specific story; we want to generate viable and novel stories. We focus on measuring both the fluency of our models and their ability to adhere to the prompt. For automatic evaluation, we measure model perplexity on the test set and prompt ranking accu racy. Perplexity is commonly used to evaluate the quality of language models, and it reflects how flu ently the model can produce the correct next word given the preceding words. We use prompt rank ing to assess how strongly a model’s output de pends on its input. Stories are decoded under 10 different prompts—9 randomly sampled prompts and 1 true corresponding prompt—and the like lihood of the story given the various prompts is recorded. We measure the percentage of cases where the true prompt is the most likely to gen erate the story.

[REF2] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking/BIBREF8_47354d4d1915ae3d286d401005ba8a44af7d1fa5.pdf  Title: A Deep Look into Neural Ranking Models for Information Retrieval  Chunk of text: Since 2016, the study of neural ranking models has bloomed, with significant work volume, deeper and more rigorous discussions, and much wider applications . For example, re searchers began to discuss the practical effectiveness of neural ranking models on different ranking tasks [21, 22]. Neural ranking models have been applied to ad-hoc retrieval [23, 24], community-based QA , conversational search , and so on. Researchers began to go beyond the architecture of neural ranking models, paying attention to new training paradigms of neural ranking models , alternate indexing schemes for neural representations , integration of 3external knowledge [29, 30], and other novel uses of neural approaches for IR tasks [31, 32]. Up to now, we have seen exciting progress on neural ranking models. In academia, several neural ranking models learned from scratch can already out perform state-of-the-art LTR models with tens of hand-crafted features [33, 34].

[REF3] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Document_expansion_learning/BIBREF79_1225eb6570ce8d45067329fafcc8ff7636a65923.pdf  Title: Modeling and Solving Term Mismatch for Full-Text Retrieval  Chunk of text: The Modern Information Retrieval System There are many challenges involved in making a retrieval system successful. These challenges include ac quiring lots of documents from many sources, estimating the quality of the acquired documents, extracting effective representations of the

documents to facilitate search and other applications, ranking documents in response to a user request, presenting search results effectively, and all other efforts involved in tracking and analyzing user behavior and search engine performance. The ranking problem is the central problem in a search engine, where all available information from the user, the search request and the document collection are used to determine the ranking of the result documents. To solve search ranking, researchers often start with simplified models, which are commonly referred to as retrieval models. Even though modern retrieval systems typically use a multitude of features for ranking documents, such as document quality and popularity estimates, user browsing and searching behavior, and other contextual information, the backbone for search ranking is usually still the standard probabilistic retrieval models such as Okapi BM25 (Robertson et al. 1995) or Statistical Language Models (Ponte and Croft 1998; Zhai and Lafferty 2001). Current retrieval models typically use simple collection statistics to assess the importance of a query term and to score and rank result documents. Most of these models are based on the tf and idf statistics, where tf, short for term frequency, is the occurrence frequency of a term in a document, and idf, the inverse document frequency, is the inverse of the occurrence frequency of a term in the whole document collection.

[REF4] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Sparse_representation_learning/BIBREF95_1e8a6de5561f557ff9abf43d538d8d5e9347efa0.pdf Title: SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking Chunk of text: On a set of 10k documents, the SPLADE-ℓFLOPS from Table 1 drops in average 20 terms per docu ment, while adding 32 expansion terms. For one of our most efficient model (FLOPS=0.05), 34 terms are dropped in average, for only 5 new expansion terms. In this case, representations are extremely sparse: documents and queries contain in average 18 and 6 non-zero values respectively, and we need less that 1.4 GB to store the index on disk. Table 2 shows an example where the model performs term re-weighting by emphasizing on important terms and discarding most of the terms without information content. Expansion allows to enrich documents, either by implicitly adding stemming effects (legs → leg) or by adding relevant topic words (e.g. treatment). 5 CONCLUSION Recently, dense retrieval based on BERT has demonstrated its supe riority for first-stage retrieval, questioning the competitiveness of traditional sparse models. In this work, we have proposed SPLADE, a sparse model revisiting document expansion.

[REF5] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Document_expansion_learning/BIBREF79_1225eb6570ce8d45067329fafcc8ff7636a65923.pdf Title: Modeling and Solving Term Mismatch for Full-Text Retrieval Chunk of text: To our knowledge, this is the first work to report performance from applying true recall weights on retrieval models other than BIM. Improvements over state-of-the-art models underscore the potential of applying recall prediction. Table 6.1: Retrieval performance with true recall weighted query terms, in Mean Average Precision. Bold face means significant by both randomization and sign tests with significance level p< 0.05. Queries are generated from TREC description query fields. TREC

dataset Document collection Topic numbers LM desc Baseline LM desc P(t|R) Improve ment p - ran domization p - sign test 4 disk 2,3 201-250 0.1789 0.2703 51.09% 0.0000 0.0000 6 disk 4,5 301-350 0.1586 0.2808 77.05% 0.0000 0.0000 8 d4,5-cr 401-450 0.1923 0.3057 58.97% 0.0000 0.0000 9 WT10g 451-500 0.2145 0.2774 29.32% 0.0000 0.0005 10 501-550 0.1627 0.2271 39.58% 0.0000 0.0000 12 .GOV

[REF6] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Document_expansion_learning/BIBREF79_1225eb6570ce8d45067329fafcc8ff7636a65923.pdf Title: Modeling and Solving Term Mismatch for Full-Text Retrieval Chunk of text: The traditional transfer learning approach aims to discover effective parameter functions that map training set statistics to model parameters, so that functions such as tf.idf that rely on heuristics and en gineering may be discovered automatically. In the context of information retrieval, there is no existing parameter function for the prediction of effective term weights or P(t|R), and such a function needs to be learned. Features that correlate well with effective term weights or with the term recall probabilities are used for predicting the target values. These features need to be general to adapt to different queries and dif ferent query terms in order to transfer knowledge about effective term weights or term recall probabilities from the training sets to the test tasks (queries). Some of these example features include the occurrence frequency of the query term in a query log (Bendersky and Croft 2008), the number of synonyms a query term has and the likelihood of a query term's synonyms appearing in place of the original query term in the collection (Zhao and Callan 2010). 2.6 Retrieval Techniques for Solving Term Mismatch In the information retrieval literature, there are many different techniques that addresses term mismatch in one way or another. Since the dissertation research is about term mismatch, we summarize related prior techniques.

[REF7] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Document_expansion_learning/BIBREF79_1225eb6570ce8d45067329fafcc8ff7636a65923.pdf Title: Modeling and Solving Term Mismatch for Full-Text Retrieval Chunk of text: That's why librarians, experts that know the index system well, are introduced to help the naive user. Ever since the capability and speed of modern computers grew to be advanced enough to allow indexing of full text, full text indexing has been the norm, sometimes even replacing the manual labeling and indexing of docu ments. The switch into full text indexing significantly reduced the chances of query term mismatch, and the retrieval systems have been much more user friendly. Similarly, when documents are short, e.g. tweets, mismatch is much more likely to happen. Thus, Twitter users insert or append hash codes to their tweets to facilitate search and reduce mismatch for commonly mentioned concepts. These hash codes behave just like the index terms in the library system but are created and used by the users instead of librarians. Inlink Anchor for Hypertext Documents Full text indexing is still not enough for solving mismatch.

[REF8] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking/BIBREF8_47354d4d1915ae3d286d401005ba8a44af7d1fa5.pdf Title: A Deep Look into Neural Ranking Models for Information Retrieval Chunk

of text: Hybrid Architecture: In order to take advantage of both representation focused and interaction-focused architectures, a natural way is to adopt a hybrid architecture for feature learning. We find that there are two major hybrid strategies to integrate the two architectures, namely combined strategy and coupled strategy. • Combined strategy is a loose hybrid strategy, which simply adopts both representation-focused and interaction-focused architectures as sub-models and combines their outputs for final relevance estimation. A representa tive model using this strategy is DUET .

[REF9] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Document_expansion_learning/BIBREF79_1225eb6570ce8d45067329fafcc8ff7636a65923.pdf  Title: Modeling and Solving Term Mismatch for Full-Text Retrieval  Chunk of text: When using the Relevance Model weights as the only feature, max-normalization is around 3% better on two datasets (TREC 10 and 12) and around 0.5% worse on two other datasets (TREC 6 and 8) than sum-normalization. (The two schemes perform similarly on TREC 4 and 14 datasets.) The difference between the normalization schemes be come slightly larger when combining the Relevance Model weight feature with the other features, with max-normalization around 3% better on TREC 4, 8, 10 and sum-normalization only better on TREC 6. There are two reasons max-normalization may be better than sum-normalization. The first reason is that the sum becomes larger as the number of query terms in a query increases, causing feature values after normalization to be smaller than those from short queries. This bias is not completely justified. Although the average term recall probability can decrease slightly as the query gets longer, there can still be query terms in long queries that have high recall probabilities.

.....
.....
.....