Title: Text Representations for Ranking - BOW Encodings

Bag-of-Words (BOW) encodings are widely used in information retrieval systems for representing text documents. In this encoding scheme, each document is represented as a vector, where each dimension corresponds to a unique term in the vocabulary. The value in each dimension represents the frequency or presence of the corresponding term in the document [REF0].

The BOW encoding approach has been shown to perform well in various subject areas and has not been surpassed by other indexing and analysis procedures [REF0]. One advantage of BOW encodings is their simplicity and efficiency in capturing the overall content of a document. By considering the frequency or presence of terms, BOW encodings can capture important information about the document's topic and relevance.

One important consideration in BOW encodings is the treatment of term frequency. The frequency of a term in a document can provide valuable information about its importance. However, it is also important to avoid overemphasizing the contribution of highly frequent terms, which may not necessarily be discriminative [REF2]. Saturation, a property of the BM25 weighting function, limits the contribution of a term to the document score, preventing highly frequent terms from dominating the ranking process [REF2].

Another aspect to consider in BOW encodings is the impact of term reweighting and query expansion. While term reweighting based on relevance feedback can improve search performance, it is not always effective in practice [REF4]. Additionally, expanding the query by adding new terms can also enhance retrieval effectiveness [REF4]. However, these techniques come with costs, such as the need for human evaluation and the computational complexity of the optimization process [REF5].

In BOW encodings, the choice of terms to include in the encoding can significantly impact retrieval performance. Discriminative terms, which have medium document frequency, are often used for content identification without further transformation [REF6]. On the other hand, highly frequent terms can be transformed into lower frequency entities, such as indexing phrases, to improve retrieval performance [REF6].

While BOW encodings have been widely used, alternative approaches have also been proposed. For instance, probabilistic models integrate indexing and retrieval models, using collection statistics to estimate the probabilities of assigning concepts to documents [REF7]. Language models, on the other hand, focus on estimating the probability of generating the query according to each document's language model [REF8].

In conclusion, BOW encodings provide a simple and efficient representation for ranking text documents in information retrieval systems. By considering term frequency and making appropriate adjustments, BOW encodings can capture important information about document relevance. However, alternative approaches, such as probabilistic models and language models, offer different perspectives on text representation and retrieval effectiveness [REF7] [REF8].

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-BOW\_Encodings/BIBREF9\_d5f169880e30e1f76827d72f862555d00b01bed9.pdf Title: A Vector Space Model for Automatic Indexing Chunk of text: A conclusive proof relating the space density analysis and the resulting document frequency indexing model to optimality in the retrieval performance cannot be furnished. However, the model appears to perform well for collections in several different subject areas, and the performance results produced by applying the theory have not in the authors' experience been surpassed by any other manual or automatic indexing and analysis procedures tried in earlier experiments. The model may then lead to the best performance obtainable with ordinary document collections operating in actual user environments. Received July 1974; revised Marcia 1975

[REF1] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-BOW\_Encodings/BIBREF9\_d5f169880e30e1f76827d72f862555d00b01bed9.pdf Title: A Vector Space Model for Automatic Indexing Chunk of text: When recall is plotted against precision, the curve closest to the upper right-hand corner of the graph (where both recall and precision are close to 1) reflects the best performance. It may be seen from Figure 8 that the replacement of the high frequency nondiscriminators by lower frequency phrases improves the retrieval performance by an average of 39 percent (the precision values at the ten fixed recall points are greater by an average of 39 percent). The performance of the right-to-left (phrase) transformation and left-to-right (thesaurus) transformation is summarized in Table IV for the three previously mentioned test collections. The precision values obtainable are near 90 percent for low recall, between 40 and 70 percent for medium recall, and between 15 and 45 percent at the high recall end of the performance spectrum. The overall improvement obtainable by phrase and thesaurus class assignments over the standard term frequency process using only the unmodified single terms ranges from 18 percent for the world affairs collection to 50 percent for the medical collection. A conclusive proof relating the space density analysis and the resulting document frequency indexing model to optimality in the retrieval performance cannot be furnished. However, the model appears to perform well for collections in several different subject areas, and the performance results produced by applying the theory have not in the authors' experience been surpassed by any other manual or automatic indexing and analysis procedures tried in earlier experiments.

[REF2] - 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 refer to this behaviour as saturation. That is, any one term’s contribution to the document score cannot exceed a saturation point (the asymptotic limit), however, frequently it occurs in the document. This turns out to be a very valuable property of the BM25 weighting function defined below. 3.4.3 A Special Case There is one case in which the saturation limit does not apply. If we assume that the eliteness property for each query term coincides with relevance for the query/need, so that pi1 = 1 and pi0 = 0, then the limit is infinite, and the weight becomes linear in tf . Thus the commonly used term-weighting functions such as the traditional tf \*idf , linear in tf , seem to fit with such a model. However, the non-linear, saturating function of tf developed below (also combined with an idf component) has frequently been shown to work better than traditional tf \*idf .3.4

[REF3] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-BOW\_Encodings/BIBREF9\_d5f169880e30e1f76827d72f862555d00b01bed9.pdf Title: A Vector Space Model for Automatic Indexing Chunk of text: I I I I I I I

[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: If we start with no relevance information, then we would weight the terms using the inverse document frequency (IDF) formula. Once the user makes some judgements of relevance, we should clearly reweight the terms according to the RSJ formula. But term reweighting is not in general an effective method for improving search. Additionally, we have to consider expanding the query by adding new terms. At an early stage in the development of the basic model, rather than considering the entire vocabulary of terms in the estimation of350 Derived Models probabilities, we restricted ourselves to query terms only. This was not because we assumed that non-query terms were incapable of giving us any useful information, but rather that in the absence of any evidence about either which terms might be useful, or how useful they might be, a reasonable neutral prior assumption was that all non-query terms had zero correlation with relevance. However, in the relevance feedback scenario discussed above, we do indeed have some evidence for the inclusion of non-query terms.

[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: Nevertheless significant gains in relevance can be obtained by properly optimising the parameters, specially when we deal with a new collection. Parameter optimisation comes with considerable costs: it will require the human evaluation of many query results, which is expensive, and the optimised parameters will be specific to the collection evaluated and may not work well for other collections. Furthermore, the optimisation procedure can be computationally costly, requiring more computing power that the search engine itself. For these reasons this approach is only appropriate for specific collections which merit the cost needed to optimise the ranking function. Examples of such collections are the WWW, large corporate collections or high-value News or Help sites. Let us call θ the vector of all free parameters of the ranking model being tuned. In the case of BM25 this vector would have two 377378 Parameter Optimisation components: θ = (k1, b).

[REF6] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-BOW\_Encodings/BIBREF9\_d5f169880e30e1f76827d72f862555d00b01bed9.pdf Title: A Vector Space Model for Automatic Indexing Chunk of text: The 70 percent of the terms with the lowest document frequency are generally poor discriminators. The best discriminators are the 25 percent whose document frequency lies approximately between 17/100 and tl/10 for n documents. If the model of Figure 7 is a correct representation of the situation relating to term importance, the following indexing strategy results [6, 7]: (a) Terms with medium document frequency should be used for content identification directly, without further transformation. (b) Terms with very high document frequency should be moved to the left on the document frequency spectrum by transforming them into entities of lower frequency; the best way of doing this is by taking highfrequency terms and using them as components of indexing phrases--a phrase such as "programming language" will necessarily exhibit lower document Fig. 7. Summarization of discrimination value of terms in frequency ranges. left -to-

[REF7] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-BOW\_Encodings/BIBREF11\_73a76dd71abfbd29dbba4ea034ab52284626aa71.pdf Title: A Language Modeling Approach to Information Retrieval Chunk of text: An additional probabilistic model is that of Fuhr . A notable feature of the Fuhr model is the integration of indexing and retrieval models. The main difference between this approach and ours is that in the Fuhr model the collection statistics are used in a heuristic fashion in order to estimate the probabilities of assigning concepts to documents. In our approach, we are able to avoid USing heuristic methods since we are not inferring concepts from terms. Another recent probabilistic approach is the INQUERY inference network model by Turtle and Croft .

[REF8] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-BOW\_Encodings/BIBREF11\_73a76dd71abfbd29dbba4ea034ab52284626aa71.pdf Title: A Language Modeling Approach to Information Retrieval Chunk of text: In the context of the retrieval task, we can treat the generation of queries as a random process. Generally speaking, language models for speech attempt to predict the probability of the next word in an ordered sequence. For the purposes of document retrieval, one can model occurrences at the document level without regard to sequential effects and will be the approach taken here. It is also possible to model local predictive effects for features such as phrases but that will be left for future work. Regarding query generation as a random process, it is not the case that queries really are generated randomly, but it is the case that retrieval systems are not endowed with knowledge of the generation process. Instead, we will treat language generation as a random process modeled by a probability distribution and focus on the estimation of probabilities as a means of achieving effective retrieval. Our approach to retrieval is to infer a language model for each document and to estimate the probability of generating the query according to each of these models.

[REF9] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-BOW\_Encodings/BIBREF5\_3cf0822f63e51be5343028bad7ee72a5882ef7de.pdf Title: Scalability Challenges in Web Search Engines Chunk of text: This repository is converted into an index by another computer that runs an indexer . The query processor evaluates a query sequentially over the constructed web index . In this architecture, users issue their queries directly to the query processing node and receive the search results from this node. 5.1.1 Single-Node Crawling A standard sequential crawler works as follows. It starts with a given set of seed URLs and iteratively fetches these URLs from the Web by establishing HTTP connections with their web servers. Downloaded pages are stored inSingle node Multi-node cluster Multi-cluster site Multi-site engine Crawling Indexing Query processing web partitioning link exchange web repartitioning collection selection index partitioning load balancing document clustering tier selection focused crawling index pruning tiering full index replication caching personalization early termination query degradation mirror detection politeness link farm detection spider trap detection multi-threading DNS caching data structures document id reassignment duplicate elimination index compression index maintenance index creation partial index replication query forwarding search site placement query routing crawler placement web partitioning search bot detection Figure 2: Issues that have an impact on scalability. a repository (see for repository management issues).

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

Title: Text Representations for Ranking - LTR Features

In the context of Neural Information Retrieval, text representations play a crucial role in ranking documents for a given query. These representations capture the semantic and contextual information of the text, enabling effective ranking algorithms. In this section, we discuss the use of Learning to Rank (LTR) features for text representations in the ranking process.

LTR features are derived from the training phase, where query/document pairs are labeled for relevance. These labels indicate the degree of match between a document and a query, such as "excellent match" or "good match" [REF0]. Unlike traditional ranking approaches that consider a single set of objects to be ranked, LTR features partition the data by query, allowing for more fine-grained ranking.

One popular approach for LTR features is LambdaRank, which optimizes ranking quality measures like Normalized Discounted Cumulative Gain (NDCG) [REF1]. LambdaRank estimates the gradient of NDCG by fixing all weights except one and computing the variation in NDCG. By plotting the NDCG as a function of the weights, LambdaRank identifies the weights that maximize NDCG, indicating the vanishing gradient at the learned values [REF1].

Another important aspect of LTR features is the consideration of different information retrieval measures. Measures like Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), Expected Reciprocal Rank (ERR), and NDCG handle multiple levels of relevance and incorporate position dependence for results shown to the user [REF3]. These measures provide a comprehensive evaluation of the ranking quality and guide the design of LTR features.

The choice of document representation for generating LTR features is also crucial. Different representations may be suitable for different types of information needs. For example, considering anchor text in the document representation may be more effective for navigational queries, while it may reduce the identification of relevant documents for informational queries [REF5]. Therefore, the type of information need observed in the queries and the document representation used for generating samples can impact the effectiveness of learned models [REF5].

To summarize, LTR features provide a powerful approach for text representations in the ranking process. LambdaRank optimization enables the estimation of gradients for ranking quality measures like NDCG, ensuring effective ranking [REF1]. Additionally, considering different information retrieval measures and choosing appropriate document representations contribute to the overall effectiveness of LTR features [REF3] [REF5].

[REF0] Introduction

[REF1] LambdaRank: Empirical Optimization of NDCG (or other IR Measures)

[REF3] Information Retrieval Measures

[REF5] The observed effectiveness of learned models can be affected by the type of information need observed in the queries, and the used document representation for generating the samples, regardless of the size of these samples.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-LTR\_Features/BIBREF16\_63aaf12163fe9735dfe9a69114937c4fa34f303a.pdf Title: Learning to Rank using Gradient Descent Chunk of text: Introduction Any system that presents results to a user, ordered by a utility function that the user cares about, is performing a ranking function. A common example is the ranking of search results, for example from the Web or from an intranet; this is the task we will consider in this paper. For this problem, the data consists of a set of queries, and for each query, a set of returned documents. In the training phase, some query/document pairs are labeled for relevance (“excellent match”, “good match”, etc.). Only those documents returned for a given query are to be ranked against each other. Thus, rather than consisting of a single set of objects to be ranked amongst each other, the data is instead partitioned by query. In this paper we propose a new approach to this problem.

[REF1] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-LTR\_Features/BIBREF15\_0df9c70875783a73ce1e933079f328e8cf5e9ea2.pdf Title: From RankNet to LambdaRank to LabdaMART: An Overview Chunk of text: 4.2 LambdaRank: Empirical Optimization of NDCG (or other IR Measures) Here we briefly describe how we showed empirically that LambdaRank directly optimizes NDCG [12, 7]. Suppose that we have trained a model and that its (learned) parameter values are w ∗ k . We can estimate a smoothed version of the gradient empirically by fixing all weights but one (call it wi), computing how the NDCG (averaged over a large number of training queries) varies, and forming the ratio δM δwi = M −M∗ wi −w ∗ i where for n queries M ≡ 1 n n ∑ i=1 NDCG(i) (8) and where the ith query has NDCG equal to NDCG(i). Now suppose we plot M as a function of wi for each i. If we observe that M is a maximum at wi = w ∗ i for every i, then we know that the function has vanishing gradient at the learned values of the weights, w = w ∗ . (Of course, if we zoom in on the graph with sufficient magnification, we’ll find that the curves are little step functions; we are considering the gradient at a scale at which the curves are smooth). This is necessary but not sufficient to show that the NDCG is a maximum at w = w ∗ : it could be a saddle point.

[REF2] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-LTR\_Features/BIBREF15\_0df9c70875783a73ce1e933079f328e8cf5e9ea2.pdf Title: From RankNet to LambdaRank to LabdaMART: An Overview Chunk of text: Note that this does not mean that the gradients are not gradients of a cost. In this section, for concreteness we assume that we are designing a model to learn NDCG.8 Christopher J.C. Burges Microsoft Research Technical Report MSR-TR-2010-82 4.1 From RankNet to LambdaRank The key observation of LambdaRank is thus that in order to train a model, we don’t need the costs themselves: we only need the gradients (of the costs with respect to the model scores). The arrows (λ’s) mentioned above are exactly those gradients. The λ’s for a given URL U1 get contributions from all other URLs for the same query that have different labels. The λ’s can also be interpreted as forces (which are gradients of a potential function, when the forces are conservative): if U2 is more relevant than U1, then U1 will get a push downwards of size |λ| (and U2, an equal and opposite push upwards); if U2 is less relevant than U1, then U1 will get a push upwards of size |λ| (and U2, an equal and opposite push downwards). Experiments have shown that modifying Eq. (3) by simply multiplying by the size of the change in NDCG (|∆NDCG|) given by swapping the rank positions of U1 and U2 (while leaving the rank positions of all other urls unchanged) gives very good results .

[REF3] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-LTR\_Features/BIBREF15\_0df9c70875783a73ce1e933079f328e8cf5e9ea2.pdf Title: From RankNet to LambdaRank to LabdaMART: An Overview Chunk of text: This led to a very significant speedup in RankNet training (since a weight update is expensive, since e.g. for a neural net model, it requires a backprop). In fact training time dropped from close to quadratic in the number of urls per query, to close to linear. It also laid the groundwork for LambdaRank, but before we discuss that, let’s review the information retrieval measures we wish to learn. 3 Information Retrieval Measures Information retrieval researchers use ranking quality measures such as Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), Expected Reciprocal Rank (ERR), and Normalized Discounted Cumulative Gain (NDCG). NDCG and ERR have the advantage that they handle multiple levels of relevance (whereas MRR and MAP are designed for binary relevance levels), and that the measure includes a position dependence for results shown to the user (that gives higher ranked results more weight), which is particularly appropriate for web search.

[REF4] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-LTR\_Features/BIBREF15\_0df9c70875783a73ce1e933079f328e8cf5e9ea2.pdf Title: From RankNet to LambdaRank to LabdaMART: An Overview Chunk of text: For the two leaf nodes of our stump, a value γl , l = 1,2 is computed, which is just the mean of the y’s of the samples that fall there. In a general regression tree, this process is continued L−1 times to form a tree with L leaves3 . 3 We are overloading notation (the meaning of L) in just this paragraph.12 Christopher J.C. Burges Microsoft Research Technical Report MSR-TR-2010-82 MART is a class of boosting algorithms that may be viewed as performing gradient descent in function space, using regression trees. The final model again maps an input feature vector x ∈ R d to a score F(x) ∈ R. MART is a class of algorithms, rather than a single algorithm, because it can be trained to minimize general costs (to solve, for example, classification, regression or ranking problems). Note, however, that the underlying model upon which MART is built is the least squares regression tree, whatever problem MART is solving. MART’s output F(x) can be written as FN(x)

[REF5] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-LTR\_Features/BIBREF12\_008f1d2741ebef51e6400686b050e046455b52fb.pdf Title: The Whens and Hows of Learning to Rank for Web Search Chunk of text: Moreover, the most suitable document representation for sampling may vary across different types of information needs. For instance, if the document representation used by the weighting model for obtaining the sample does consider anchor text, then the navigational pages with quality anchor text in their incoming hyperlinks are more likely to be retrieved in the sample (Hawking et al 2004; Plachouras and Ounis 2004). However, we postulate that using anchor text may reduce the number of relevant documents identified for more informational queries. Hence, we hypothesise that: Hypothesis 2 The observed effectiveness of learned models can be affected by the type of information need observed in the queries, and the used document representation for generating the samples, regardless of the size of these samples. Moreover, the choice of learning to rank technique may also have an impact on the effective choice of the sample size.

[REF6] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-LTR\_Features/BIBREF15\_0df9c70875783a73ce1e933079f328e8cf5e9ea2.pdf Title: From RankNet to LambdaRank to LabdaMART: An Overview Chunk of text: Algorithm: LambdaMART set number of trees N, number of training samples m, number of leaves per tree L, learning rate η for i = 0 to m do F0(xi) = BaseModel(xi) //If BaseModel is empty, set F0(xi) = 0 end for for k = 1 to N do for i = 0 to m do yi = λi wi = ∂ yi ∂Fk−1 (xi) end for {Rlk} L l=1 // Create L leaf tree on {xi ,yi} m i=1 γlk

[REF7] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-LTR\_Features/BIBREF12\_008f1d2741ebef51e6400686b050e046455b52fb.pdf Title: The Whens and Hows of Learning to Rank for Web Search Chunk of text: .037 .220 .154 TD04 MAP .182 .004 .220 .017 .220 .019 .175 .002 .151 .050 WT09 ERR@20 .136 .000 .158 .012 .141 .009 .133 .001 .122 .022

[REF8] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-LTR\_Features/BIBREF16\_63aaf12163fe9735dfe9a69114937c4fa34f303a.pdf Title: Learning to Rank using Gradient Descent Chunk of text: Unlabeled documents were given rating 0. Ranking accuracy was computed using a normalized discounted cumulative gain measure (NDCG) (Jarvelin & Kekalainen, 2000). We chose to compute the NDCG at rank 15, a little beyond the set of documents initially viewed by most users. For a given query qi , the results are sorted by decreasing score output by the algorithm, and the NDCG is then computed as Ni ≡ Ni X 15 j=1 (2r(j) − 1)/ log(1 + j) (16) where r(j) is the rating of the j’th document, and where the normalization constant Ni is chosen so that a perfect ordering gets NDCG score 1. For those queries with fewer than 15 returned documents, the NDCG was computed for all the returned documents. Note that unlabeled documents does not contribute to the sum directly, but will still reduce the NDCG by displacing labeled documents; also note that Ni = 1 is an unlikely event, even for a perfect ranker, since some unlabeled documents may in fact be highly relevant.

[REF9] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-LTR\_Features/BIBREF16\_63aaf12163fe9735dfe9a69114937c4fa34f303a.pdf Title: Learning to Rank using Gradient Descent Chunk of text: [A, B] in Rd , together with target probabilities P¯AB that sample A is to be ranked higher than sample B. This is a general formulation: the pairs of ranks need not be complete (inLearning to Rank using Gradient Descent that taken together, they need not specify a complete ranking of the training data), or even consistent. We consider models f : Rd 7→ R such that the rank order of a set of test samples is specified by the real values that f takes, specifically, f(x1) > f(x2) is taken to mean that the model asserts that x1 B x2. Denote the modeled posterior P(xi Bxj ) by Pij , i, j = 1, . . . , m, and let P¯ ij be the desired target values for those posteriors. Define oi ≡ f(xi) and oij ≡ f(xi) − f(xj ). We will use the cross entropy cost function Cij ≡ C(oij )

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

Title: Text Representations for Ranking - Word Embeddings

Word embeddings have emerged as a powerful technique for representing text in neural information retrieval systems. Word embeddings capture the semantic and syntactic relationships between words by mapping them to dense vector representations in a continuous space. These representations have been widely used in various natural language processing tasks, including information retrieval and ranking.

One popular approach for generating word embeddings is the use of pre-trained models such as GloVe [REF2] and word2vec [REF2]. These models learn word embeddings by leveraging the distributional properties of words in large corpora. GloVe, for example, constructs word embeddings based on the co-occurrence statistics of words in a corpus [REF4]. It captures the global patterns of word co-occurrence, which allows it to capture different types of semantic relationships [REF2]. On the other hand, word2vec is trained to predict words within a local context window, resulting in embeddings that capture more local relationships [REF2].

Word embeddings can be used to represent both query and document text in neural information retrieval systems. The embeddings of the query and document words are combined to form a representation of the query-document pair, which is then used for ranking. Various methods have been proposed to combine word embeddings, including simple averaging, weighted averaging, and more sophisticated neural network architectures [REF0].

One advantage of using word embeddings for ranking is their ability to capture semantic similarity between words. Words that are semantically similar tend to have similar vector representations in the embedding space. This allows the retrieval system to effectively match queries with relevant documents based on their semantic content. Additionally, word embeddings can capture syntactic relationships between words, such as word order and grammatical structure, which can be useful for ranking documents based on their syntactic similarity to the query.

To further enhance the performance of word embeddings for ranking, several techniques have been proposed. One approach is to fine-tune the pre-trained word embeddings on a specific ranking task [REF0]. This involves training a neural network model on a large labeled dataset, where the word embeddings are updated during the training process. Fine-tuning helps to adapt the word embeddings to the specific ranking task, improving their effectiveness in capturing the relevance between queries and documents [REF1].

Another technique is to incorporate language modeling as an auxiliary objective during the fine-tuning process [REF1]. Language modeling helps to improve the generalization of the supervised model and accelerate convergence. By optimizing a combined objective that includes both the ranking loss and the language modeling loss, the fine-tuned model can better capture the semantic and syntactic properties of the text, leading to improved ranking performance [REF1].

In conclusion, word embeddings have become a popular choice for representing text in neural information retrieval systems. They capture semantic and syntactic relationships between words, allowing for effective matching of queries and documents. Fine-tuning and incorporating language modeling as auxiliary objectives further enhance the performance of word embeddings for ranking. These techniques have shown promising results in various information retrieval tasks and continue to be an active area of research in neural information retrieval [REF9].

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-Word\_Embeddings/BIBREF23\_cd18800a0fe0b668a1cc19f2ec95b5003d0a5035.pdf Title: Improving Language Understanding by Generative Pre-Training Chunk of text: 3Figure 1: (left) Transformer architecture and training objectives used in this work. (right) Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer. 3.3 Task-specific input transformations For some tasks, like text classification, we can directly fine-tune our model as described above. Certain other tasks, like question answering or textual entailment, have structured inputs such as ordered sentence pairs, or triplets of document, question, and answers. Since our pre-trained model was trained on contiguous sequences of text, we require some modifications to apply it to these tasks. Previous work proposed learning task specific architectures on top of transferred representations .

[REF1] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-Word\_Embeddings/BIBREF23\_cd18800a0fe0b668a1cc19f2ec95b5003d0a5035.pdf Title: Improving Language Understanding by Generative Pre-Training Chunk of text: (4) We additionally found that including language modeling as an auxiliary objective to the fine-tuning helped learning by (a) improving generalization of the supervised model, and (b) accelerating convergence. This is in line with prior work [50, 43], who also observed improved performance with such an auxiliary objective. Specifically, we optimize the following objective (with weight λ): L3(C) = L2(C) + λ ∗ L1(C) (5) Overall, the only extra parameters we require during fine-tuning are Wy, and embeddings for delimiter tokens (described below in Section 3.3). 3Figure 1: (left) Transformer architecture and training objectives used in this work. (right) Input transformations for fine-tuning on different tasks.

[REF2] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-Word\_Embeddings/BIBREF17\_5303f288c0de1fc717c3389773a2a684589ee46b.pdf Title: Semantic Memory Search and Retrieval in a Novel Cooperative Word Game: A Comparison of Associative and Distributional Semantic Models Chunk of text: Within this context, associative models may provide an important behavioral baseline or benchmark for comparisons across DSMs and may therefore be useful in assessing 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. Although error-driven distributional models such as word2vec have been shown to outperform error-free learning models in other psycholinguistic tasks (e.g., Mandera et al., 2017), it is possible that GloVe may be more sensitive to different types of semantic relationships due to capturing more global patterns of co-occurrence, compared to word2vec, which is trained to predict words within a local context window. This may have contributed to the better performance of GloVe in the speaker task within the game. However, word2vec and GloVe appeared to perform at similar levels in the guesser task and in accounting for the response latency patterns, suggesting that these models may also share similar mechanisms to some degree (see Levy & Goldberg, 2014). Given that pretrained models were used in this study, future work should look into how different hyperparameters 15516709, 2021, 10, Downloaded from <https://onlinelibrary.wiley.com/doi/10.1111/cogs.13053> by University Estadual De Campina, Wiley Online Library on [17/06/2023]. See the Terms and Conditions (<https://onlinelibrary.wiley.com/terms-and-conditions>) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License28 of 33 A. A. Kumar, M. Steyvers, D. A. Balota / Cognitive Science 45 (2021) influence the generated semantic representations to better understand the relative performance of different distributional models, as well as explore more advanced language models.

[REF3] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-Word\_Embeddings/BIBREF21\_df2b0e26d0599ce3e70df8a9da02e51594e0e992.pdf Title: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Chunk of text: Intuitively, it is reasonable to believe that a deep bidirectional model is strictly more powerful than either a left-to-right model or the shallow concatenation of a left-toright and a right-to-left model. Unfortunately, standard conditional language models can only be trained left-to-right or right-to-left, since bidirectional conditioning would allow each word to indirectly “see itself”, and the model could trivially predict the target word in a multi-layered context. former is often referred to as a “Transformer encoder” while the left-context-only version is referred to as a “Transformer decoder” since it can be used for text generation. In order to train a deep bidirectional representation, we simply mask some percentage of the input tokens at random, and then predict those masked tokens. We refer to this procedure as a “masked LM” (MLM), although it is often referred to as a Cloze task in the literature (Taylor, 1953). In this case, the final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary, as in a standard LM. In all of our experiments, we mask 15% of all WordPiece tokens in each sequence at random.

[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: 10−5 P(k|ice)/P(k|steam) 8.9 8.5 × 10−2 1.36 0.96 context of word i. We begin with a simple example that showcases how certain aspects of meaning can be extracted directly from co-occurrence probabilities. Consider two words i and j that exhibit a particular aspect of interest; for concreteness, suppose we are interested in the concept of thermodynamic phase, for which we might take i = ice and j = steam. The relationship of these words can be examined by studying the ratio of their co-occurrence probabilities with various probe words, k. For words k related to ice but not steam, say k = solid, we expect the ratio Pik /Pjk will be large.

[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: i w˜ k ) = Pik = Xik Xi . (5) The solution to Eqn. (4) is F = exp, or, w T i w˜ k = log(Pik ) = log(Xik ) − log(Xi) .

[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: Training proceeds in an on-line, stochastic fashion, but the implied global objective function can be written as, J = − X i∈corpus j∈context(i) log Qi j . (11) Evaluating the normalization factor of the softmax for each term in this sum is costly. To allow for efficient training, the skip-gram and ivLBL models introduce approximations to Qi j. However, the sum in Eqn. (11) can be evaluated muchmore efficiently if we first group together those terms that have the same values for i and j, J = − X V i=1 X V j=1 Xi j log Qi j , (12) where we have used the fact that the number of like terms is given by the co-occurrence matrix X. Recalling our notation for Xi = P k Xik and Pi j = Xi j/Xi , we can rewrite J as, J = − X V i=1 Xi X V j=1 Pi j log Qi j = X V i=1 XiH(Pi ,Qi) , (13) where H(Pi ,Qi) is the cross entropy of the distributions Pi and Qi , which we define in analogy to Xi . As a weighted sum of cross-entropy error, this objective bears some formal resemblance to the weighted least squares objective of Eqn.

[REF7] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-Word\_Embeddings/BIBREF20\_f37e1b62a767a307c046404ca96bc140b3e68cb5.pdf Title: GloVe: Global Vectors for Word Representation Chunk of text: Though we did not do so, this step could easily be parallelized across multiple machines (see, e.g., Lebret and Collobert (2014) for some benchmarks). Using a single thread of a dual 2.1GHz Intel Xeon E5-2658 machine, populating X with a 10 word symmetric context window, a 400,000 word vocabulary, and a 6 billion token corpus takes about 85 minutes. Given X, the time it takes to train the model depends on the vector size and the number of iterations. For 300-dimensional vectors with the above settings (and using all 32 cores of the above machine), a single iteration takes 14 minutes. See Fig. 4 for a plot of the learning curve. 4.7 Model Analysis: Comparison with word2vec A rigorous quantitative comparison of GloVe with word2vec is complicated by the existence of many parameters that have a strong effect on performance.

[REF8] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-Word\_Embeddings/BIBREF18\_87f40e6f3022adbc1f1905e3e506abad05a9964f.pdf Title: Distributed Representations of Words and Phrases and their Compositionality Chunk of text: Four closest tokens to the sum of two vectors are shown, using the best Skip-gram model. To maximize the accuracy on the phrase analogy task, we increased the amount of the training data by using a dataset with about 33 billion words. We used the hierarchical softmax, dimensionality of 1000, and the entire sentence for the context. This resulted in a model that reached an accuracy of 72%. We achieved lower accuracy 66% when we reduced the size of the training dataset to 6B words, which suggests that the large amount of the training data is crucial. To gain further insight into how different the representations learned by different models are, we did inspect manually the nearest neighbours of infrequent phrases using various models. In Table 4, we show a sample of such comparison.

[REF9] - paperID: ./papers\_pdf/paper\_section/Text\_Representations\_for\_Ranking-Word\_Embeddings/BIBREF22\_077f8329a7b6fa3b7c877a57b81eb6c18b5f87de.pdf Title: RoBERTa: A Robustly Optimized BERT Pretraining Approach Chunk of text: These results depend on a several task-specific modifications, which we describe in Section 5.1. SQuAD The Stanford Question Answering Dataset (SQuAD) provides a paragraph of context and a question. The task is to answer the question by extracting the relevant span from the context. We evaluate on two versions of SQuAD: V1.1 and V2.0 (Rajpurkar et al., 2016, 2018). In V1.1 the context always contains an answer, whereas in 5The authors and their affiliated institutions are not in any way affiliated with the creation of the OpenWebText dataset. 6The datasets are: CoLA (Warstadt et al., 2018), Stanford Sentiment Treebank (SST) (Socher et al., 2013), Microsoft Research Paragraph Corpus (MRPC) (Dolan and Brockett, 2005), Semantic Textual Similarity Benchmark (STS) (Agirre et al., 2007), Quora Question Pairs (QQP) (Iyer et al., 2016), MultiGenre NLI (MNLI) (Williams et al., 2018), Question NLI (QNLI) (Rajpurkar et al., 2016), Recognizing Textual Entailment (RTE) (Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009) and Winograd NLI (WNLI) (Levesque et al., 2011).V2.0 some questions are not answered in the provided context, making the task more challenging.

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

Title: Interaction-focused Systems - Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have been widely used in interaction-focused systems for neural information retrieval. These systems aim to capture the hierarchical matching patterns between query and document pairs to improve relevance matching. In this section, we discuss the use of CNNs in interaction-focused systems and highlight the key differences between these systems and traditional retrieval models.

Existing interaction-focused models, such as ARC-II and MatchPyramid, employ CNNs to learn hierarchical matching patterns over the matching matrix [REF1]. These models use convolutional units with a local "receptive field" to capture positional regularities in matching patterns. However, this approach may not be suitable for ad-hoc retrieval tasks, as relevance matching in these tasks does not necessarily exhibit positional regularity [REF1]. Additionally, CNN parameters in these models treat both exact matching and similarity matching signals equally [REF1]. In contrast, interaction-focused systems using CNNs for ad-hoc retrieval aim to extract hierarchical matching patterns from different levels of interaction signals rather than different positions [REF1].

One example of an interaction-focused system is the Deep Relevance Matching Model (DRMM) [REF0]. DRMM employs a fixed-length matching histogram to capture matching patterns and a feed-forward matching network to learn hierarchical patterns [REF0]. The model also includes a term gating network to compute aggregation weights for the matching scores of each query term [REF0]. Experimental results show that DRMM outperforms traditional retrieval models and existing deep matching models on benchmark collections [REF0].

The design of interaction-focused systems is influenced by the differences between relevance matching in ad-hoc retrieval and semantic matching in other natural language processing tasks [REF3]. Relevance matching in ad-hoc retrieval may be global, assuming short documents have a concentrated topic, or it may occur in any part of a long document due to the Scope Hypothesis [REF3]. These differences affect the architecture of deep models, making it challenging to find a universal solution for different matching problems [REF3]. Most existing deep matching models focus on semantic matching rather than relevance matching, emphasizing compositional meaning and global matching requirements [REF3].

In future work, it is suggested to leverage larger training data, such as click-through logs, to train deeper models like DRMM and explore their potential in ad-hoc retrieval [REF4]. Additionally, incorporating phrase embeddings to treat phrases as a whole rather than separate terms may improve retrieval performance by better reflecting the meaning through local interactions [REF4].

In conclusion, CNNs play a crucial role in interaction-focused systems for neural information retrieval. These systems aim to capture hierarchical matching patterns and address the diverse requirements of relevance matching in ad-hoc retrieval tasks. By leveraging CNNs and considering the differences between relevance matching and semantic matching, these systems have shown promising results in improving retrieval performance.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Convolutional\_Neural\_Networks/BIBREF25\_d51ed05fd05b9d222427a05a87ed88217447b44f.pdf Title: A Deep Relevance Matching Model for Ad-hoc Retrieval Chunk of text: Based on this fixed-length matching histogram, we then employ a feed forward matching network to learn hierarchical matching patterns and produce a matching score. Finally, the overall matching score is generated by aggregating the scores from each query term with a term gating network computing the aggregation weights. We show how our major model designs, including matching histogram mapping, a feed forward matching network, and a term gating network, address the three key factors in relevance matching for ad-hoc retrieval. We evaluate the effectiveness of the proposed DRMM based on two representative ad-hoc retrieval benchmark collections. For comparison, we take into account some wellknown traditional retrieval models, as well as several stateof-the-art deep matching models either designed for the general matching problem or proposed specifically for the adhoc retrieval task. The empirical results show that the existing deep matching models cannot compete with the traditional retrieval models on these benchmark collections, while our model can outperform all the baseline models significantly in terms of all the evaluation metrics. The major contributions of this paper include: 1.

[REF1] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Convolutional\_Neural\_Networks/BIBREF25\_d51ed05fd05b9d222427a05a87ed88217447b44f.pdf Title: A Deep Relevance Matching Model for Ad-hoc Retrieval Chunk of text: Since our model follows the approach of interaction-focused models, we discuss the major differences between the learning of our feed forward matching network and that in previous interactionfocused models. Existing interaction-focused models, e.g., ARC-II and MatchPyramid, employ a CNN to learn hierarchical matching patterns over the matching matrix. These models are basically position-aware using convolutional units with a local “receptive field” and learning positional regularities in matching patterns. This may be suitable for the image recognition task, and work well on semantic matching problems due to the global matching requirement (i.e., all the positions are important). However, it may not be suitable for the ad-hoc retrieval task, since such positional regularity may not exist in relevance matching due to the diverse matching requirement discussed in Section 3. Besides, since CNN parameters are position related, these models will treat both exact matching and similarity matching signals equally. Our deep relevance matching model, on the contrary, aims to extract hierarchical matching patterns from different levels of interaction signals rather than different positions.

[REF2] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Convolutional\_Neural\_Networks/BIBREF27\_fc3384d631f5e2b2a9d66623d4d3e1d28b96dee7.pdf Title: Convolutional Neural Networks for Soft-Machting N-Grams in Ad-hoc Search Chunk of text: Since this evaluation uses sessions with only one click, the MRR scores directly reect the reciprocal rank of userclicked documents. On Sogou-Log, the average rank of clicked documents of all methods except K-NRM and Conv-KNRM was below rank 5. K-NRM pulled the clicked document to rank 3, and Conv-KNRM further promoted it to rank 2.7. On Bing-Log, Conv-KNRM pulled the clicked document of all methods more than 1 position higher. e only neural IR baselines that outperformed feature-based learning-to-rank are the two interaction based and end-to-end trained ones: MP and K-NRM. Although other neural IR methods can improve over unsupervised baselines, feature-based learning-to-rank methods are harder to beat; end-to-end learned embeddings and matchbased techniques are necessary for current neural IR methods to provide additional improvements [22, 29, 30] Comparing the two strong neural IR baselines, K-NRM outperforms MP by a large margin.

[REF3] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Convolutional\_Neural\_Networks/BIBREF25\_d51ed05fd05b9d222427a05a87ed88217447b44f.pdf Title: A Deep Relevance Matching Model for Ad-hoc Retrieval Chunk of text: In this case, the relevance matching might be global if we assume short documents have a concentrated topic. On the contrary, the Scope Hypothesis assumes a long document consists of a number of unrelated short documents concatenated together. In this way, the relevance matching could happen in any part of a relevant document, and we do not require the document as a whole to be relevant to a query. As we can see, there are significant differences between relevance matching in ad-hoc retrieval and semantic matching in many NLP tasks. These differences affect the design of deep model architectures and it may be difficult to find a “one-fit-all” solution to such different matching problems. If we revisit the existing deep matching models, we find that most of them concern semantic matching rather than relevance matching. For example, the representation-focused models such as DSSM, C-DSSM and ARC-I focus on the compositional meaning of the texts and fit the global matching requirement.

[REF4] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Convolutional\_Neural\_Networks/BIBREF25\_d51ed05fd05b9d222427a05a87ed88217447b44f.pdf Title: A Deep Relevance Matching Model for Ad-hoc Retrieval Chunk of text: For future work, we would like to leverage larger training data, e.g. click-through logs, to train deeper DRMM so that we can further explore the potential of the proposed model on ad-hoc retrieval. We may also include phrase embeddings so that phrases can be treated as a whole rather than separate terms. In this way, we expect the local interactions can better reflect the meaning by using the proper semantic units in language, leading to better retrieval performance. 8. ACKNOWLEDGMENTS This work was supported in part by the Center for Intelligent Information Retrieval, in part by the 973 Program of China under Grant No. 2014CB340401 and 2013CB329606, in part by the National Natural Science Foundation of China under Grant No. 61232010, 61472401, 61425016, and 61203298, and in part by the Youth Innovation Promotion Association CAS under Grant No. 20144310 and 2016102. 9.

[REF5] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Convolutional\_Neural\_Networks/BIBREF26\_ea738439b880ad033ff01602ea52d04b366d0d37.pdf Title: End-to-End Neural Ad-hod Ranking with Kernel Pooling Chunk of text: . At query time, the ranking function considers the similarities of all query and document word pairs, allowing query words to be so-matched to document words. e translation matrix can be calculated via mutual information in a corpus or using user clicks . Word pair interactions have also been modeled by word embeddings. Word embeddings trained from surrounding contexts, for example, word2vec , are considered to be the factorization of word pairs’ PMI matrix

[REF6] - 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: The input plane receives images of characters that are approximately size-normalized and centered. Each unit of a layer receives inputs from a set of units located in a small neighborhood in the previous layer. The idea of connecting units to local receptive elds on the input goes back to the Perceptron in the early 60s, and was almost simultaneous with Hubel and Wiesel's discovery of locallysensitive, orientation-selective neurons in the cat's visual system. Local connections have been reused many times in neural models of visual learning (see (Mozer, 1991; Le Cun, 1986) and NEOCOGNITRON in this handbook). With local receptive elds, neurons can extract elementary visual features such as oriented edges, end-points, corners (or similar features in speech spectrograms). These features are then combined by the higher layers. As stated earlier, distortions or shifts of the input can cause the position of salient features to vary.

[REF7] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Convolutional\_Neural\_Networks/BIBREF26\_ea738439b880ad033ff01602ea52d04b366d0d37.pdf Title: End-to-End Neural Ad-hod Ranking with Kernel Pooling Chunk of text: CDSSM learns hundreds of convolution lters on Chinese characters, thus has millions of parameters. K-NRM’s parameter space is even larger as it learns an embedding for every Chinese word. Models with more parameters in general are expected to t beer but may also require more training data to avoid overing. 5 EVALUATION RESULTS Our experiments investigated K-NRM’s eectiveness, as well as its behavior on tail queries, with less training data, and with dierent kernel widths. Table 4: Ranking accuracy of K-NRM and baseline methods. Relative performances compared with Coor-Ascent are in percentages. Win/Tie/Loss are the number of queries improved, unchanged, or hurt, compared to Coor-Ascent on NDCG@10.

[REF8] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Convolutional\_Neural\_Networks/BIBREF27\_fc3384d631f5e2b2a9d66623d4d3e1d28b96dee7.pdf Title: Convolutional Neural Networks for Soft-Machting N-Grams in Ad-hoc Search Chunk of text: K-NRM is a state-of-the-art neural model previously tested on the Sogou-Log dataset . It uses kernel-pooling instead of DRMM’s histogram pooling, and learns the word embeddings and the ranking layers end-to-end. It is the main baseline in our experiments. Among these neural IR baselines, DRMM and K-NRM were compared on the ClueWeb09-B dataset. DRMM uses xed embeddings and only learns the learning-to-rank layers, and can be trained with limited training data. K-NRM was tested the same as Conv-KNRM in the domain adaption fashion. MP and CDSSM performed worse than DRMM on TREC data in previous studies [13, 22].

[REF9] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Convolutional\_Neural\_Networks/BIBREF26\_ea738439b880ad033ff01602ea52d04b366d0d37.pdf Title: End-to-End Neural Ad-hod Ranking with Kernel Pooling Chunk of text: Note that there was no overlap between training queries and testing queries. Testing-SAME infers relevance labels using DCTR, the same click model used for training. is seing evaluates the ranking model’s ability to t user preferences (click through rates). Testing-DIFF infers relevance scores using TACM , a stateof-the-art click model. TACM is a more sophisticated model and uses both clicks and dwell times.

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

Title: Interaction-focused Systems - Pre-trained Language Models

Pre-trained language models have revolutionized the field of neural information retrieval by providing a powerful framework for understanding and generating text. These models leverage transfer learning, treating every text processing problem as a "text-to-text" problem, where text is taken as input and new text is generated as output [REF0]. This approach allows for the application of the same model, objective, training procedure, and decoding process to various tasks, enabling flexibility and efficiency in handling different NLP problems [REF0].

One prominent example of pre-trained language models is the Transformer architecture, which has gained significant attention in recent years. The Transformer utilizes self-attention mechanisms to capture dependencies between words in a sentence, enabling it to effectively model long-range dependencies [REF1]. The self-attention mechanism in the decoder also employs autoregressive or causal self-attention, restricting the model to attend only to past outputs [REF1]. Additionally, relative position embeddings have been introduced to provide explicit position signals to the Transformer, enhancing its ability to understand the order of words in a sequence [REF1].

To evaluate the performance of pre-trained language models, various benchmark datasets have been utilized. These datasets include tasks such as Definite Pronoun Resolution (DPR), CNN/Daily Mail for text summarization, and SQuAD for question answering [REF2]. By training on these datasets, pre-trained language models can learn to perform well on a wide range of NLP problems.

In the context of neural information retrieval, pre-trained language models have been applied in interaction-focused systems. These systems aim to enhance the interaction between users and retrieval systems by incorporating user feedback and adapting the retrieval process accordingly. One approach is to use pre-trained language models to generate query suggestions based on user input, improving the relevance and effectiveness of search queries [REF3]. Another approach is to utilize pre-trained language models to personalize search results by considering user preferences and past interactions [REF4].

The training of pre-trained language models involves various strategies, such as proportional mixing, temperature-scaled mixing, and equal mixing of data sets [REF5]. These strategies determine the proportions of data from different tasks that the model is trained on, ensuring that the model sees enough data to perform well on each task without overfitting or underfitting [REF5].

Comparative studies have shown that the BERT-style objective, which involves reconstructing the original uncorrupted text segment, performs well in pre-training language models [REF6]. However, alternative objectives such as prefix language modeling and deshuffling have been explored, with varying degrees of success [REF6]. The choice of objective can significantly impact the performance of pre-trained language models in neural information retrieval tasks.

Efficiency and generalization are important considerations in the training of pre-trained language models. Techniques such as distillation loss and cosine embedding loss have been employed to improve efficiency and align the hidden state vectors of student and teacher models [REF8]. Additionally, the impact of tasks such as Next Sentence Prediction (NSP) and bidirectional representations on model performance has been studied, highlighting the importance of these factors in achieving strong results [REF9].

In summary, pre-trained language models have emerged as a powerful tool in interaction-focused systems for neural information retrieval. These models leverage transfer learning, utilize the Transformer architecture, and are trained on benchmark datasets to perform well on a wide range of NLP tasks. The choice of training objectives, mixing strategies, and efficiency considerations play crucial roles in the performance and applicability of pre-trained language models in the field of neural information retrieval.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Pre-trained\_Language\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: Motivated by a need for more rigorous understanding, we leverage a unified approach to transfer learning that allows us to systematically study different approaches and push the current limits of the field. The basic idea underlying our work is to treat every text processing problem as a “text-to-text” problem, i.e. taking text as input and producing new text as output. This approach is inspired by previous unifying frameworks for NLP tasks, including casting all text problems as question answering (McCann et al., 2018), language modeling (Radford et al., 2019), or span extraction Keskar et al. (2019b) tasks. Crucially, the text-to-text framework allows us to directly apply the same model, objective, training procedure, and decoding process to every task we consider. We leverage this flexibility by evaluating performance on a wide variety of English-based NLP problems, including question answering, document 2. [http://commoncrawl.org](http://commoncrawl.org/) 2Exploring the Limits of Transfer Learning "translate English to German: That is good." "cola sentence: The course is jumping well."

[REF1] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Pre-trained\_Language\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: <http://nlp.seas.harvard.edu/2018/04/03/attention.html> 4. <http://jalammar.github.io/illustrated-transformer/> 4Exploring the Limits of Transfer Learning mechanism after each self-attention layer that attends to the output of the encoder. The self-attention mechanism in the decoder also uses a form of autoregressive or causal selfattention, which only allows the model to attend to past outputs. The output of the final decoder block is fed into a dense layer with a softmax output, whose weights are shared with the input embedding matrix. All attention mechanisms in the Transformer are split up into independent “heads” whose outputs are concatenated before being further processed. Since self-attention is order-independent (i.e. it is an operation on sets), it is common to provide an explicit position signal to the Transformer. While the original Transformer used a sinusoidal position signal or learned position embeddings, it has recently become more common to use relative position embeddings (Shaw et al., 2018; Huang et al., 2018a). Instead of using a fixed embedding for each position, relative position embeddings produce a different learned embedding according to the offset between the “key” and “query” being compared in the self-attention mechanism.

[REF2] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Pre-trained\_Language\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: As suggested by Kocijan et al. (2019) we also include the Definite Pronoun Resolution (DPR) data set (Rahman and Ng, 2012) in the combined SuperGLUE task. The CNN/Daily Mail (Hermann et al., 2015) data set was introduced as a questionanswering task but was adapted for text summarization by Nallapati et al. (2016) ; we use the non-anonymized version from See et al. (2017) as an abstractive summarization task. SQuAD (Rajpurkar et al., 2016) is a common question-answering benchmark.

[REF3] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Pre-trained\_Language\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: To feed this example into a language model, we would transform it into the sequence “mnli premise: I hate pigeons. hypothesis: My feelings towards pigeons are filled with animosity. target: entailment”. In this case, the fully-visible prefix would correspond to the entire input sequence up to the word “target:”, which can be seen as being analogous to the “classification” token used in BERT. So, our model would have full visibility over the entire input, and then would be tasked with making a classification by outputting the word “entailment”. It is easy for the model to learn to output one of the valid class labels given the task prefix (“mnli” in this case). As such, the main difference between a prefix LM and the BERT architecture is that the classifier is simply integrated into the output layer of the Transformer decoder in the prefix LM.

[REF4] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Pre-trained\_Language\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: Examplesproportional mixing refers to sampling examples from each data set according to the total size of each data set, with an artificial limit (K) on the maximum data set size. Temperature-scaled mixing re-scales the sampling rates by a temperature T. For temperature-scaled mixing, we use an artificial data set size limit of K = 221 . Equal mixing In this case, we sample examples from each task with equal probability. Specifically, each example in each batch is sampled uniformly at random from one of the data sets we train on. This is most likely a suboptimal strategy, as the model will overfit quickly on low-resource tasks and underfit on high-resource tasks. We mainly include it as a point of reference of what might go wrong when the proportions are set suboptimally. To compare these mixing strategies on equal footing with our baseline pre-train-thenfine-tune results, we train multi-task models for the same total number of steps: 2 19 + 218 = 786,432.

[REF5] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Pre-trained\_Language\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: For example, we might train a single model on many tasks, but when reporting performance we are allowed to select a different checkpoint for each task. This loosens the multi-task learning framework and puts it on more even footing compared to the pre-train-then-fine-tune approach we have considered so far. We also note that in our 30Exploring the Limits of Transfer Learning unified text-to-text framework, “multi-task learning” simply corresponds to mixing data sets together. It follows that we can still train on unlabeled data when using multi-task learning by treating the unsupervised task as one of the tasks being mixed together. In contrast, most applications of multi-task learning to NLP add task-specific classification networks or use different loss functions for each task (Liu et al., 2019b). As pointed out by Arivazhagan et al. (2019), an extremely important factor in multi-task learning is how much data from each task the model should be trained on. Our goal is to not under- or over-train the model—that is, we want the model to see enough data from a given task that it can perform the task well, but not to see so much data that it memorizes the training set.

[REF6] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Pre-trained\_Language\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: Overall, we find that the BERT-style objective performs best, though the prefix language modeling objective attains similar performance on the translation tasks. Indeed, the motivation for the BERT objective was to outperform language model-based pre-training. The deshuffling objective performs considerably worse than both prefix language modeling and the BERT-style objective. 21Raffel, Shazeer, Roberts, Lee, Narang, Matena, Zhou, Li and Liu Objective GLUE CNNDM SQuAD SGLUE EnDe EnFr EnRo BERT-style (Devlin et al., 2018) 82.96 19.17 80.65 69.85 26.78 40.03 27.41 MASS-style (Song et al., 2019) 82.32 19.16 80.10 69.28 26.79 39.89 27.55 F Replace corrupted spans 83.28 19.24 80.88 71.36 26.98 39.82 27.65 Drop corrupted tokens 84.44 19.31 80.52 68.67 27.07 39.76 27.82 Table 5: Comparison of variants of the BERT-style pre-training objective. In the first two variants, the model is trained to reconstruct the original uncorrupted text segment.

[REF7] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Pre-trained\_Language\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: We suspect that this simplistic technique may not be a very efficient way to teach the model general-purpose knowledge. More concretely, it would be useful to be able to attain good fine-tuning performance without needing to train our models on 1 trillion tokens of text first. Some concurrent work along these lines improves efficiency by pre-training a model to distinguish between real and machine-generated text (Clark et al., 2020). Formalizing the similarity between tasks We observed that pre-training on unlabeled in-domain data can improve performance on downstream tasks (Section 3.4). This finding mostly relies on basic observations like the fact that SQuAD was created using data from Wikipedia. It would be useful to formulate a more rigorous notion of the “similarity” between the pre-training and downstream tasks, so that we could make more principled choices about what source of unlabeled data to use. There is some early empirical work along these lines in the field of computer vision (Huh et al., 2016; Kornblith et al., 2018; He et al., 2018).

[REF8] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Pre-trained\_Language\_Models/BIBREF34\_a54b56af24bb4873ed0163b77df63b92bd018ddc.pdf Title: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter Chunk of text: The final training objective is a linear combination of the distillation loss Lce with the supervised training loss, in our case the masked language modeling loss Lmlm [Devlin et al., 2018]. We found it beneficial to add a cosine embedding loss (Lcos) which will tend to align the directions of the student and teacher hidden states vectors. 3 DistilBERT: a distilled version of BERT Student architecture In the present work, the student - DistilBERT - has the same general architecture as BERT. The token-type embeddings and the pooler are removed while the number of layers is reduced by a factor of 2. Most of the operations used in the Transformer architecture (linear layer and layer normalisation) are highly optimized in modern linear algebra frameworks and our investigations showed that variations on the last dimension of the tensor (hidden size dimension) have a smaller impact on computation efficiency (for a fixed parameters budget) than variations on other factors like the number of layers. Thus we focus on reducing the number of layers.

[REF9] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Pre-trained\_Language\_Models/BIBREF21\_df2b0e26d0599ce3e70df8a9da02e51594e0e992.pdf Title: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Chunk of text: This is directly comparable to OpenAI GPT, but using our larger training dataset, our input representation, and our fine-tuning scheme. We first examine the impact brought by the NSP task. In Table 5, we show that removing NSP hurts performance significantly on QNLI, MNLI, and SQuAD 1.1. Next, we evaluate the impact of training bidirectional representations by comparing “No NSP” to “LTR & No NSP”. The LTR model performs worse than the MLM model on all tasks, with large drops on MRPC and SQuAD. For SQuAD it is intuitively clear that a LTR model will perform poorly at token predictions, since the token-level hidden states have no rightside context. In order to make a good faith attempt at strengthening the LTR system, we added a randomly initialized BiLSTM on top.

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

Title: Interaction-focused Systems - Ranking with Encoder-only Models

Encoder-only models have gained significant attention in the field of neural information retrieval due to their ability to effectively rank documents based on user interactions. These models focus on capturing the relevance between queries and documents by leveraging the power of deep learning techniques.

One approach to ranking with encoder-only models is based on the use of conditional probability. By decomposing the conditional probability of a sequence, the relevance of a document given a query can be estimated [REF0]. This approach considers the probability of each symbol in the target sequence given the source sentence encoding and the decoded target sequence so far. The decoder, implemented as a combination of an RNN network and a softmax layer, generates a probability distribution over candidate output symbols [REF0].

Another challenge in ranking with encoder-only models is the impact of quantization errors, especially when dealing with deep LSTMs and long sequences. To address this challenge, techniques such as quantized arithmetic can be employed to speed up inference without compromising translation quality [REF1]. Additional constraints can be added during training to ensure that the model is quantizable with minimal impact on the output. Experimental results have shown that these constraints do not hinder model convergence or the quality of the model [REF1].

Passage re-ranking is an essential stage in interaction-focused systems, where each document is scored and re-ranked using a more computationally-intensive method [REF2]. BERT, a popular language model, can be used as a re-ranker by feeding the query as sentence A and the passage text as sentence B [REF2]. By truncating the query to a certain number of tokens, the relevance score of each candidate passage can be estimated [REF2].

The advent of deep learning has revolutionized the field of information retrieval, particularly in the context of neural ranking models [REF3]. These models leverage continuous vector space representations and neural architectures to eliminate the need for manual feature engineering [REF3]. Various neural ranking models, such as DRMM, DUET, KNRM, and Co-PACRR, have been proposed and have shown promising results [REF3]. It is important to note that most neural ranking models are re-ranking models, operating over a list of candidate documents generated by a query [REF3].

Incorporating contextualized word representations into existing neural architectures has been shown to improve ad-hoc document ranking [REF4]. Joint models that combine BERT's classification vector with existing neural ranking architectures have been proposed to leverage the benefits of both approaches [REF4]. Additionally, techniques for addressing the performance impact of computing contextualized language models have been explored, such as partial computation of language model representations [REF4].

Efficiency and effectiveness are crucial factors in interaction-focused systems. The effectiveness of duoBERT, a combination of monoBERT and duoBERT stages, has been evaluated by varying the number of candidates [REF5]. Aggregation methods like BINARY and SUM have shown promising results in terms of effectiveness on different datasets [REF5]. The number of inferences per query can be optimized based on the number of candidates, providing a trade-off between efficiency and effectiveness [REF5].

Relevance matching models have proven to be effective in neural ranking, without requiring massive amounts of behavioral data [REF6]. Techniques such as PACRR increase the maximum number of term matches to improve back-propagation to the language model [REF6]. Contextualized language models, such as ELMo and BERT, have been widely used in neural ranking experiments [REF6]. While ELMo encodes the query and document separately, BERT enables simultaneous encoding of the query and document using Segment A and Segment B embeddings [REF6].

In the design and implementation of encoder-only models, the use of LSTM RNNs with residual connections has shown promising results [REF7]. Residual connections between layers encourage gradient flow and improve the accuracy of the models [REF7]. To enhance parallelism, attention is connected from the bottom layer of the decoder network to the top layer of the encoder network [REF8]. Low-precision arithmetic and specialized hardware, such as Google's Tensor Processing Unit (TPU), are employed to improve inference time [REF8]. Sub-word units, known as "wordpieces," are used to effectively handle rare words [REF8].

To rank candidate translations, a scoring function that incorporates length normalization and coverage penalty can be employed [REF9]. Length normalization ensures that the length of the translation does not bias the ranking, while the coverage penalty favors translations that fully cover the source sentence [REF9]. The scoring function takes into account the conditional probability of the translation given the source sentence and the coverage of the source sentence by the translation [REF9].

In summary, interaction-focused systems that utilize encoder-only models have shown promising results in ranking documents based on user interactions. These models leverage techniques such as conditional probability, quantized arithmetic, and contextualized language models to improve the accuracy and efficiency of the ranking process. The design and implementation of these models involve considerations such as residual connections, low-precision arithmetic, and scoring functions that account for length normalization and coverage penalty.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-only\_Models/BIBREF37\_dbde7dfa6cae81df8ac19ef500c42db96c3d1edd.pdf Title: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation Chunk of text: Using the chain rule the conditional probability of the sequence P(Y |X) can be decomposed as: P(Y |X) = P(Y |x1, x2, x3, ..., xM) = Y N i=1 P(yi |y0, y1, y2, ..., yi−1; x1, x2, x3, ..., xM) (2) where y0 is a special “beginning of sentence” symbol that is prepended to every target sentence. During inference we calculate the probability of the next symbol given the source sentence encoding and the decoded target sequence so far: P(yi |y0, y1, y2, y3, ..., yi−1; x1, x2, x3, ..., xM) (3) Our decoder is implemented as a combination of an RNN network and a softmax layer. The decoder RNN network produces a hidden state yi for the next symbol to be predicted, which then goes through the softmax layer to generate a probability distribution over candidate output symbols. In our experiments we found that for NMT systems to achieve good accuracy, both the encoder and decoder RNNs have to be deep enough to capture subtle irregularities in the source and target languages. This observation is similar to previous observations that deep LSTMs significantly outperform shallow LSTMs .

[REF1] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-only\_Models/BIBREF37\_dbde7dfa6cae81df8ac19ef500c42db96c3d1edd.pdf Title: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation Chunk of text: Deep LSTMs with long sequences pose a novel challenge in that quantization errors can be significantly amplified after many unrolled steps or after going through a deep LSTM stack. In this section, we present our approach to speed up inference with quantized arithmetic. Our solution is tailored towards the hardware options available at Google. To reduce quantization errors, additional constraints are added to our model during training so that it is quantizable with minimal impact on the output of the model. That is, once a model is trained with these additional constraints, it can be subsequently quantized without loss to translation quality. Our experimental results suggest that those additional constraints do not hurt model convergence nor the quality of a model once it has converged. Recall from equation 6 that in an LSTM stack with residual connections there are two accumulators: c

[REF2] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-only\_Models/BIBREF38\_85e07116316e686bf787114ba10ca60f4ea7c5b2.pdf Title: Passage Re-Ranking with BERT Chunk of text: In the second stage, passage re-ranking, each of these documents is scored and re-ranked by a more computationally-intensive method. Finally, the top ten or fifty of these documents will be the source for the candidate answers by an answer generation module. In this paper, we describe how we implemented the second stage of this pipeline, passage re-ranking. Method The job of the re-ranker is to estimate a score si of how relevant a candidate passage di is to a query q. We use BERT as our re-ranker. Using the same notation used by Devlin et al. 1 arXiv:1901.04085v5 [cs.IR] 14 Apr 2020(2018), we feed the query as sentence A and the passage text as sentence B. We truncate the query to have at most 64 tokens.

[REF3] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-only\_Models/BIBREF40\_63a2fabbe4b1615a84d5f4d90987733cf09e3ff8.pdf Title: Multi-Stage Document Ranking with BERT Chunk of text: In our work, we make the connection between BERTbased models and multi-stage ranking, which allows us to trade off the quality of the results with inference latency. The advent of deep learning has brought tremendous excitement into the information retrieval community. Although machine-learned ranking models have been well studied since the mid-2000s under the banner of “learning to rank”, the paradigm is heavily driven by manual feature engineering (Liu, 2009; Li, 2011); commercial web search engines are known to incorporate thousands of features (or more) in their models. Continuous vector space representations coupled with neural models promise to obviate the need for handcrafted features and have attracted the attention of many researchers. Well-known neural ranking models include DRMM (Guo et al., 2016), DUET (Mitra et al., 2017), KNRM (Xiong et al., 2017), and Co-PACRR (Hui et al., 2018); the literature is too vast for an exhaustive review here, and thus we refer readers to recent overviews (Onal et al., 2018; Mitra and Craswell, 2019). Although often glossed over, most neural ranking models today (including all the models referenced above) are actually re-ranking models, in the sense that they operate over the output of a list of candidate documents, typically produced by a “bag of words” query.

[REF4] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-only\_Models/BIBREF39\_1ec78c0ec945572673fabd50bf263870fe9d3601.pdf Title: CEDR: Contextualized Embeddings for Document Reranking Chunk of text: In summary, our contributions are as follows: - We are the first to demonstrate that contextualized word representations can be successfully incorporated into existing neural architectures (PACRR , KNRM , and DRMM ), allowing them to leverage contextual information to improve ad-hoc document ranking. - We present a new joint model that combines BERT’s classification vector with existing neural ranking architectures (using BERT’s token vectors) to get the benefits from both approaches. - We demonstrate an approach for addressing the performance impact of computing contextualized language models by only partially computing the language model representations. - Our code is available for replication and future work.1 2 METHODOLOGY 2.1 Notation In ad-hoc ranking, documents are ranked for a given query according to a relevance estimate.

[REF5] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-only\_Models/BIBREF40\_63a2fabbe4b1615a84d5f4d90987733cf09e3ff8.pdf Title: Multi-Stage Document Ranking with BERT Chunk of text: The leftmost point, k1 = 0, corresponds to monoBERT only, which allows us to quantify the additive impact of the duoBERT stage on top monoBERT results. The values for the SAMPLE method represent the average of ten trials. Of the four aggregation methods compared, BINARY yields the highest effectiveness on the MS MARCO dataset, albeit by a small margin over SUM. On TREC CAR, BINARY and SUM are very close, although SUM appears to be slightly better, especially at lower cutoffs. The SAMPLE0 500 1,000 2,000 3,450 33 35 37 38.5 # inferences/query MRR@10 MS MARCO 0 500 1,000 2,000 3,450 29 31 33 35 37.5 # inferences/query MAP k1 = 10 k1 = 30 k1 = 50 TREC CAR Figure 4: Number of inferences per query vs. the effectiveness of duoBERTSUM when varying the number of candidates k0 and k1. Each curve has five points that correspond to k0 = {50, 100, 200, 500, 1000}. The number of inferences per query is calculated as k0 + k1(k1 − 1).

[REF6] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-only\_Models/BIBREF39\_1ec78c0ec945572673fabd50bf263870fe9d3601.pdf Title: CEDR: Contextualized Embeddings for Document Reranking Chunk of text: . Relevance matching models have generally shown to be more effective than semantic matching models, while not requiring massive amounts of behavioral data (e.g., query logs). For PACRR, we increase kmax = 30 to allow for more term matches and better back-propagation to the language model. Contextualized language models. We use the pretrained ELMo (Original, 5.5B) and BERT (BERT-Base, Uncased) language models in our experiments. For ELMo, the query and document are encoded separately. Since BERT enables encoding multiple texts at the same time using Segment A and Segment B embeddings, we encode the query (Segment A) and document (Segment B) simultaneously.

[REF7] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-only\_Models/BIBREF37\_dbde7dfa6cae81df8ac19ef500c42db96c3d1edd.pdf Title: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation Chunk of text: context ai for the current time step is computed according to the following formulas: st = AttentionF unction(yi−1, xt) ∀t, 1 ≤ t ≤ M pt = exp(st)/ X M t=1 exp(st) ∀t, 1 ≤ t ≤ M ai = X M t=1 pt.xt (4) where AttentionF unction in our implementation is a feed forward network with one hidden layer. 3.1 Residual Connections As mentioned above, deep stacked LSTMs often give better accuracy over shallower models. However, simply stacking more layers of LSTM works only to a certain number of layers, beyond which the network becomes 4too slow and difficult to train, likely due to exploding and vanishing gradient problems [33, 22]. In our experience with large-scale translation tasks, simple stacked LSTM layers work well up to 4 layers, barely with 6 layers, and very poorly beyond 8 layers.

[REF8] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-only\_Models/BIBREF37\_dbde7dfa6cae81df8ac19ef500c42db96c3d1edd.pdf Title: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation Chunk of text: This work presents the design and implementation of GNMT, a production NMT system at Google, that aims to provide solutions to the above problems. In our implementation, the recurrent networks are Long Short-Term Memory (LSTM) RNNs [23, 17]. Our LSTM RNNs have 8 layers, with residual connections between layers to encourage gradient flow . For parallelism, we connect the attention from the bottom layer of the decoder network to the top layer of the encoder network. To improve inference time, we employ low-precision arithmetic for inference, which is further accelerated by special hardware (Google’s Tensor Processing Unit, or TPU). To effectively deal with rare words, we use sub-word units (also known as “wordpieces”)

[REF9] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-only\_Models/BIBREF37\_dbde7dfa6cae81df8ac19ef500c42db96c3d1edd.pdf Title: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation Chunk of text: We first tried to simply divide by the length to normalize. We then improved on that original heuristic by dividing by lengthα, with 0 < α < 1 where α is optimized on a development set (α ∈ [0.6 − 0.7] was usually found to be best). Eventually we designed the empirically-better scoring function below, which also includes a coverage penalty to favor translations that fully cover the source sentence according to the attention module. More concretely, the scoring function s(Y, X) that we employ to rank candidate translations is defined as follows: s(Y, X) = log(P(Y |X))/lp(Y ) + cp(X; Y ) lp(Y ) = (5 + |Y |) α (5 + 1)α cp(X; Y ) = β ∗ X |X| i=1 log(min(X |Y | j=1 pi,j , 1.0)), (14) where pi,j is the attention probability of the j-th target word yj on the i-th source word xi .

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

Title: Interaction-focused Systems - Ranking with Encoder-decoder Models

Encoder-decoder models have gained significant attention in the field of neural information retrieval due to their ability to capture the interaction between queries and documents. These models employ a two-step process, where the encoder encodes the query and document representations, and the decoder generates a relevance score or ranking based on the encoded representations. In this section, we discuss the use of encoder-decoder models for ranking in interaction-focused systems.

One important aspect in training encoder-decoder models is the choice of pre-training data. Liu et al. [REF0] observed that pre-training on a diverse dataset leads to improved performance on downstream tasks. This finding has motivated research on domain adaptation for natural language processing [REF0]. However, pre-training on a single domain often results in smaller datasets, which can limit the model's performance [REF0]. It is crucial to investigate the impact of using smaller pre-training datasets on the performance of encoder-decoder models [REF0].

Another approach to evaluate the effectiveness of encoder-decoder models is through open-domain cloze-style question answering tasks. REF2 compares the performance of BERT-large, a widely used encoder-decoder model, with the supervised DrQA model. The results show a performance gap between the two models, highlighting the potential of encoder-decoder models in information retrieval tasks [REF2]. Additionally, REF3 demonstrates that increasing the model size improves performance on the Children's Book Test, indicating the importance of model capacity in encoder-decoder systems [REF3].

To enhance the training process of encoder-decoder models, various objectives have been proposed. One such objective is the "masked language modeling" (MLM) objective, inspired by BERT [REF4]. MLM involves corrupting a portion of the tokens in a span of text and training the model to reconstruct the original tokens. In the case of encoder-decoder models, the entire uncorrupted sequence is used as the target [REF4]. Another objective, known as deshuffling, has been applied to denoising sequential autoencoders [REF4]. These objectives contribute to the overall performance of encoder-decoder models in ranking tasks.

The architecture of encoder-decoder models allows for task conditioning, where tasks, inputs, and outputs are specified as sequences of symbols [REF6]. This flexibility enables the model to infer and perform various tasks, such as translation and reading comprehension, using a single model [REF6]. The ability of encoder-decoder models to handle multiple tasks makes them versatile in information retrieval scenarios.

In terms of performance, encoder-decoder models have achieved state-of-the-art results in various benchmarks [REF7]. Larger model sizes have been found to be crucial for achieving the best performance across different tasks [REF7]. The performance of encoder-decoder models has surpassed previous state-of-the-art models in natural language inference tasks, where machine performance historically lagged behind human performance [REF7].

In conclusion, encoder-decoder models have shown promise in ranking tasks within interaction-focused systems. The choice of pre-training data, model capacity, and training objectives play significant roles in enhancing the performance of these models. Furthermore, the flexibility of encoder-decoder architectures allows for multitask learning and the ability to handle various information retrieval tasks.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-decoder\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: This is unsurprising but also unsatisfying if our goal is to pre-train a model that can rapidly adapt to language tasks from arbitrary domains. Liu et al. (2019c) also observed that pre-training on a more diverse data set yielded improvements on downstream tasks. This observation also motivates the parallel line of research on domain adaptation for natural language processing; for surveys of this field see e.g. Ruder (2019); Li (2012). A drawback to only pre-training on a single domain is that the resulting data sets are often substantially smaller. Similarly, while the WebText-like variant performed as well or better than the C4 data set in our baseline setting, the Reddit-based filtering produced a data set that was about 40× smaller than C4 despite being based on 12× more data from Common Crawl. Note, however, that in our baseline setup we only pre-train on 2 35 ≈ 34B tokens, which is only about 8 times larger than the smallest pre-training data set we consider. We investigate at what point using a smaller pre-training data sets poses an issue in the following section.

[REF1] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-decoder\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: Original target: 1 Processed target: contradiction 49Raffel, Shazeer, Roberts, Lee, Narang, Matena, Zhou, Li and Liu D.10 COPA Original input: Question: effect Premise: Political violence broke out in the nation. Choice 1: Many citizens relocated to the capitol. Choice 2: Many citizens took refuge in other territories. Processed input: copa choice1: Many citizens relocated to the capitol. choice2: Many citizens took refuge in other territories. premise: Political violence broke out in the nation.

[REF2] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-decoder\_Models/BIBREF42\_d0086b86103a620a86bc918746df0aa642e2a8a3.pdf Title: Language Models as Knowledge Bases? Chunk of text: Next we evaluate our system on opendomain cloze-style question answering and compare against the supervised DrQA model. Table 2 shows a performance gap between BERT-large and the DrQA open-domain QA system on our cloze SQuAD task. Again, note that the pretrained language model is completely unsupervised, it is not fine-tuned, and it has no access to a dedicated information retrieval system. Moreover, when comparing DrQA and BERT-large in terms of P@10, we find that gap is remarkably small (57.1 for BERT-large and 63.5 for DrQA). 6 Discussion and Conclusion We presented a systematic analysis of the factual and commonsense knowledge in publicly available pretrained language models as is and foundRelation Query Answer Generation T-Rex P19 Francesco Bartolomeo Conti was born in . Florence Rome [-1.8] , Florence

[REF3] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-decoder\_Models/BIBREF41\_9405cc0d6169988371b2755e573cc28650d14dfe.pdf Title: Language Models are Unsupervised Multask Learners Chunk of text: Performance on the Children’s Book Test as a function of model capacity. Human performance are from Bajgar et al. (2016), instead of the much lower estimates from the original paper. The Children’s Book Test (CBT) (Hill et al., 2015) was created to examine the performance of LMs on different categories of words: named entities, nouns, verbs, and prepositions. Rather than reporting perplexity as an evaluation metric, CBT reports accuracy on an automatically constructed cloze test where the task is to predict which of 10 possible choices for an omitted word is correct. Following the LM approach introduced in the original paper, we compute the probability of each choice and the rest of the sentence conditioned on this choice according to the LM, and predict the one with the highest probability. As seen in Figure 2 performance steadily improves as model size is increased and closes the majority of the gap to human performance on this test. Data overlap analysis showed one of the CBT test set books, The Jungle Book by Rudyard Kipling, is in WebText, so we report results on the validation set which has no significant overlap.

[REF4] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-decoder\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: Second, we consider an objective inspired by the “masked language modeling” (MLM) objective used in BERT (Devlin et al., 2018). MLM takes a span of text and corrupts 15% of the tokens. 90% of the corrupted tokens are replaced with a special mask token and 10% are replaced with a random token. Since BERT is an encoder-only model, its goal during pre-training is to reconstruct masked tokens at the output of the encoder. In the encoder-decoder case, we simply use the entire uncorrupted sequence as the target. Note that this differs from our baseline objective, which uses only the corrupted tokens as targets; we compare these two approaches in Section 3.3.2. Finally, we also consider a basic deshuffling objective as used e.g. in (Liu et al., 2019a) where it was applied to a denoising sequential autoencoder.

[REF5] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-decoder\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: BERT (Devlin et al., 2018) also uses a fully-visible masking pattern and appends a special “classification” token to the input. BERT’s output at the timestep corresponding to the classification token is then used to make a prediction for classifying the input sequence. 16Exploring the Limits of Transfer Learning The self-attention operations in the Transformer’s decoder use a “causal” masking pattern. When producing the ith entry of the output sequence, causal masking prevents the model from attending to the jth entry of the input sequence for j > i. This is used during training so that the model can’t “see into the future” as it produces its output. An attention matrix for this masking pattern is shown in Figure 3, middle. The decoder in an encoder-decoder Transformer is used to autoregressively produce an output sequence.

[REF6] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-decoder\_Models/BIBREF41\_9405cc0d6169988371b2755e573cc28650d14dfe.pdf Title: Language Models are Unsupervised Multask Learners Chunk of text: 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). But as exemplified in McCann et al. (2018), language provides a flexible way to specify tasks, inputs, and outputs all as a sequence of symbols. For example, a translation training example can be written as the sequence (translate to french, english text, french text). Likewise, a reading comprehension training example can be written as (answer the question, document, question, answer). McCann et al. (2018) demonstrated it was possible to train a single model, the MQAN,Language Models are Unsupervised Multitask Learners to infer and perform many different tasks on examples with this type of format. Language modeling is also able to, in principle, learn the tasks of McCann et al.

[REF7] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-decoder\_Models/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: Overall, we achieved state-of-the-art performance on 18 out of the 24 tasks we consider. As expected, our largest (11 billion parameter) model performed best among our model size variants across all tasks. Our T5-3B model variant did beat the previous state of the art in a few tasks, but scaling the model size to 11 billion parameters was the most important ingredient for achieving our best performance. We now analyze the results for each individual benchmark. We achieved a state-of-the-art average GLUE score of 90.3. Notably, our performance was substantially better than the previous state-of-the-art for the natural language inference tasks MNLI, RTE, and WNLI. RTE and WNLI are two of the tasks where machine performance has historically lagged behind human performance, which is 93.6 and 95.9 respectively (Wang et al., 2018).

[REF8] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-decoder\_Models/BIBREF42\_d0086b86103a620a86bc918746df0aa642e2a8a3.pdf Title: Language Models as Knowledge Bases? Chunk of text: Bb Bl Figure 2: Mean P@k curve for T-REx varying k. Base10 log scale for X axis. about the correct answer can be extracted from the output representation). Figure 2 shows the mean P@k curves for the considered models. For BERT, the correct object is ranked among the top ten in around 60% of the cases and among the top 100 in 80% of the cases. To further investigate why BERT achieves such strong results, we compute the Pearson correlation coefficient between the P@1 and a set of metrics that we report in Figure 3. We notice, for instance, that the number of times an object is mentioned in the training data positively correlates with performance while the same is not true for the subject of a relation. Furthermore, the log probability of a prediction is strongly positively correlated with P@1.

[REF9] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Ranking\_with\_Encoder-decoder\_Models/BIBREF42\_d0086b86103a620a86bc918746df0aa642e2a8a3.pdf Title: Language Models as Knowledge Bases? Chunk of text: In other words, assume we query for the object o of a test subjectrelation fact (s,r, o) expressed in a sentence x. If RE has extracted any triple (s 0 ,r, o 0 ) from that sen-tence x, s 0 will be linked to s and o 0 to o. In practice, this means RE can return the correct solution o if any relation instance of the right type was extracted from x, regardless of whether it has a wrong subject or object. DrQA: Chen et al. (2017) introduce DrQA, a popular system for open-domain question answering. DrQA predicts answers to natural language questions using a two step pipeline. First, a TF/IDF information retrieval step is used to find relevant articles from a large store of documents (e.g. Wikipedia). On the retrieved top k articles, a neural reading comprehension model then extracts answers. To avoid giving the language models a competitive advantage, we constrain the predictions of DrQA to single-token answers.

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

Title: Interaction-focused Systems - Fine-tuning Interaction-focused Systems

Fine-tuning interaction-focused systems is an important aspect of neural information retrieval. In this section, we discuss the concept of fine-tuning and its relevance in improving the performance of interaction-focused systems.

One approach to fine-tuning interaction-focused systems is through the use of similarity information. Wagstaff et al. proposed a promising approach for clustering with similarity information [REF0]. This approach involves searching for a clustering that puts similar pairs into the same clusters and dissimilar pairs into different clusters, based on the information provided. By incorporating similarity side-information, this method aims to align the clustering results with a user's notion of meaningful clusters. However, it is important to note that the constraints used in this approach do not generalize well to previously unseen data [REF0].

Another approach to fine-tuning interaction-focused systems is through the use of distance metrics. By learning a distance metric that respects the relationships between similar pairs of points, the performance of clustering algorithms can be improved [REF1]. This approach poses metric learning as a convex optimization problem, allowing for efficient algorithms that are free from local optima [REF1]. The learned metrics can be diagonal or full, depending on the problem at hand [REF1]. Experimental results have shown that using learned metrics leads to significantly improved performance over naive K-means clustering [REF3].

One distinguishing feature of fine-tuning interaction-focused systems is the ability to learn a full metric over the input space, rather than focusing solely on the training set [REF2]. This allows for better generalization to previously unseen data [REF2]. Additionally, fine-tuning methods can be used as a pre-processing step to enhance the performance of unsupervised algorithms such as LLE and MDS [REF2]. By incorporating the methods proposed in conjunction with existing approaches, better solutions can be achieved [REF2].

The evaluation of fine-tuned interaction-focused systems is crucial in assessing their effectiveness. Accuracy measures, such as the agreement between the clustering results and the "true" clustering, can be used to evaluate the performance [REF4]. Experimental results have shown that fine-tuned interaction-focused systems outperform traditional clustering algorithms, such as K-means and constrained K-means, in various datasets [REF7]. The use of learned diagonal or full metrics has led to significantly improved performance in most cases [REF7].

In conclusion, fine-tuning interaction-focused systems plays a vital role in improving the performance of neural information retrieval. By incorporating similarity information and learning distance metrics, these systems can better align with a user's notion of meaningful clusters and achieve improved clustering performance. The ability to generalize to previously unseen data and the potential for enhancing existing unsupervised algorithms make fine-tuning an important aspect of interaction-focused systems.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Fine-tuning\_Interaction-focused\_Systems/BIBREF44\_d1a2d203733208deda7427c8e20318334193d9d7.pdf Title: Distance metric learning, with application to clustering with side-information Chunk of text: While these methods often learn good metrics for classification, it is less clear whether they can be used to learn good, general metrics for other algorithms such as K-means, particularly if the information available is less structured than the traditional, homogeneous training sets expected by them. In the context of clustering, a promising approach was recently proposed by Wagstaff et al. for clustering with similarity information. If told that certain pairs are “similar” or “dissimilar,” they search for a clustering that puts the similar pairs into the same, and dissimilar pairs into different, clusters. This gives a way of using similarity side-information to find clusters that reflect a user’s notion of meaningful clusters. But similar to MDS and LLE, the (“instance-level”) constraints that they use do not generalize to previously unseen data whose similarity/dissimilarity to the training set is not known. We will later discuss this work in more detail, and also examine the effects of using the methods we propose in conjunction with these methods.

[REF1] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Fine-tuning\_Interaction-focused\_Systems/BIBREF44\_d1a2d203733208deda7427c8e20318334193d9d7.pdf Title: Distance metric learning, with application to clustering with side-information Chunk of text: For some problems (e.g., wine), our algorithm learns good diagonal and full metrics quickly with only a very small amount of side-information; for some others (e.g., protein), the distance metric, particularly the full metric, appears harder to learn and provides less benefit over constrained K-means. 4 Conclusions We have presented an algorithm that, given examples of similar pairs of points in , learns a distance metric that respects these relationships. Our method is based on posing metric learning as a convex optimization problem, which allowed us to derive efficient, localoptima free algorithms. We also showed examples of diagonal and full metrics learned from simple artificial examples, and demonstrated on artificial and on UCI datasets how our methods can be used to improve clustering performance.

[REF2] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Fine-tuning\_Interaction-focused\_Systems/BIBREF44\_d1a2d203733208deda7427c8e20318334193d9d7.pdf Title: Distance metric learning, with application to clustering with side-information Chunk of text: One feature distinguishing our work from these is that we will learn a full metric over the input space, rather than focusing only on (finding an embedding for) the points in the training set. Our learned metric thus generalizes more easily to previously unseen data. More importantly, methods such as LLE and MDS also suffer from the “no right answer” problem: For example, if MDS finds an embedding that fails to capture the structure important to a user, it is unclear what systematic corrective actions would be available. (Similar comments also apply to Principal Components Analysis (PCA) .) As in our motivating clustering example, the methods we propose can also be used in a pre-processing step to help any of these unsupervised algorithms to find better solutions. In the supervised learning setting, for instance nearest neighbor classification, numerous attempts have been made to define or learn either local or global metrics for classification.

[REF3] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Fine-tuning\_Interaction-focused\_Systems/BIBREF44\_d1a2d203733208deda7427c8e20318334193d9d7.pdf Title: Distance metric learning, with application to clustering with side-information Chunk of text: In each, we ran one experiment using “little” side-information ­ , and one with “much” side-information. The results are given in Figure 6.9 We see that, in almost every problem, using a learned diagonal or full metric leads to significantly improved performance over naive K-means. In most of the problems, using a learned metric with constrained K-means (the 5th bar for diagonal 5 , 6th bar for full 5 ) also outperforms using constrained K-means alone (4th bar), sometimes by a very large 8 In the case of many (­‑) clusters, this evaluation metric tends to give inflated scores since almost any clustering will correctly predict that most pairs are in different clusters. In this setting, we therefore modified the measure averaging not only I J , IL drawn uniformly at random, but from the same cluster (as determined by ! ) with chance 0.5, and from different clusters with chance 0.5, so that “matches” and “mis-matches” are given the same weight. All results reported here used K-means with multiple restarts, and are averages over at least 20 trials (except for wine, 10 trials). 9F was generated by picking a random subset of all pairs of points sharing the same class !

- ­ where ­ B is the indicator function ( ­ $ ­ , ­ \*/ $ < ). This is equivalent to the probability that for two pointsL ,! drawn randomly from the dataset, our clustering % agrees with the “true” clustering % on whetherZ and!belong to same or different clusters.8 As a simple example, consider Figure 4, which shows a clustering problem in which the “true clusters” (indicated by the different symbols/colors in the plot) are distinguished by their -coordinate, but where the data in its original space seems to cluster much better according to their # -coordinate. As shown by the accuracy scores given in the figure, both K-means and constrained K-means failed to find good clusterings. But by first learning a distance metric and then clustering according to that metric, we easily find the correct clustering separating the true clusters from each other.

[REF5] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Fine-tuning\_Interaction-focused\_Systems/BIBREF44\_d1a2d203733208deda7427c8e20318334193d9d7.pdf Title: Distance metric learning, with application to clustering with side-information Chunk of text: 50 −50 0 50 −50 0 50 x Original data y z −50 0 50 −50 0 50 −50 0 50 y x z Projected data (a) (b) 1. K-means: Accuracy = 0.4993 2. Constrained K-means: Accuracy = 0.5701 3.

[REF6] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Fine-tuning\_Interaction-focused\_Systems/BIBREF44\_d1a2d203733208deda7427c8e20318334193d9d7.pdf Title: Distance metric learning, with application to clustering with side-information Chunk of text: Figure 3 shows a similar result for a case of three clusters whose centroids differ only in the x and y directions. As we see in Figure 3(b), the learned diagonal metric correctly ignores the z direction. Interestingly, in the case of a full 5 , the algorithm finds a surprising projection of the data onto a line that still maintains the separation of the clusters well. 3.2 Application to clustering One application of our methods is “clustering with side information,” in which we learn a distance metric using similarity information, and cluster data using that metric. Specifically, suppose we are given ­ , and told that each pair Z C! ‑ ­ meansL and belong to the same cluster. We will consider four algorithms for clustering: 1. K-means using the default Euclidean metric \*,\* -\*,\* B B between points and cluster centroids to define distortion (and ignoring ­ ). 2.

[REF7] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Fine-tuning\_Interaction-focused\_Systems/BIBREF44\_d1a2d203733208deda7427c8e20318334193d9d7.pdf Title: Distance metric learning, with application to clustering with side-information Chunk of text: As shown by the accuracy scores given in the figure, both K-means and constrained K-means failed to find good clusterings. But by first learning a distance metric and then clustering according to that metric, we easily find the correct clustering separating the true clusters from each other. Figure 5 gives another example showing similar results. We also applied our methods to 9 datasets from the UC Irvine repository. Here, the “true clustering” is given by the data’s class labels. In each, we ran one experiment using “little” side-information ­ , and one with “much” side-information. The results are given in Figure 6.9 We see that, in almost every problem, using a learned diagonal or full metric leads to significantly improved performance over naive K-means.

[REF8] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Fine-tuning\_Interaction-focused\_Systems/BIBREF44\_d1a2d203733208deda7427c8e20318334193d9d7.pdf Title: Distance metric learning, with application to clustering with side-information Chunk of text: Introduction The performance of many learning and datamining algorithms depend critically on their being given a good metric over the input space. For instance, K-means, nearest-neighbors classifiers and kernel algorithmssuch as SVMs all need to be given good metrics that reflect reasonably well the important relationships between the data. This problem is particularly acute in unsupervised settings such as clustering, and is related to the perennial problem of there often being no “right” answer for clustering: If three algorithms are used to cluster a set of documents, and one clusters according to the authorship, another clusters according to topic, and a third clusters according to writing style, who is to say which is the “right” answer? Worse, if an algorithm were to have clustered by topic, and if we instead wanted it to cluster by writing style, there are relatively few systematic mechanisms for us to convey this to a clustering algorithm, and we are often left tweaking distance metrics by hand. In this paper, we are interested in the following problem: Suppose a user indicates that certain points in an input space (say, ) are considered by them to be “similar.” Can we automatically learn a distance metric over that respects these relationships, i.e., one that assigns small distances between the similar pairs? For instance, in the documents example, we might hope that, by giving it pairs of documents judged to be written in similar styles, it would learn to recognize the critical features for determining style���������One important family of algorithms that (implicitly) learn metrics are the unsupervised ones that take an input dataset, and find an embedding of it in some space.

[REF9] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Fine-tuning\_Interaction-focused\_Systems/BIBREF44\_d1a2d203733208deda7427c8e20318334193d9d7.pdf Title: Distance metric learning, with application to clustering with side-information Chunk of text: As in our motivating clustering example, the methods we propose can also be used in a pre-processing step to help any of these unsupervised algorithms to find better solutions. In the supervised learning setting, for instance nearest neighbor classification, numerous attempts have been made to define or learn either local or global metrics for classification. In these problems, a clear-cut, supervised criterion—classification error—is available and can be optimized for. (See also , for a different way of supervising clustering.) This literature is too wide to survey here, but some relevant examples include [10, 5, 3, 6], and also gives a good overview of some of this work. While these methods often learn good metrics for classification, it is less clear whether they can be used to learn good, general metrics for other algorithms such as K-means, particularly if the information available is less structured than the traditional, homogeneous training sets expected by them. In the context of clustering, a promising approach was recently proposed by Wagstaff et al.

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

Title: Interaction-focused Systems - Dealing with long texts

In the field of Neural Information Retrieval, interaction-focused systems have gained significant attention due to their ability to handle long texts effectively. Long texts, such as documents or articles, pose a challenge for traditional information retrieval systems as they require a deeper understanding of the content to provide accurate and relevant results. In this section, we discuss various approaches and techniques used in interaction-focused systems to address the issue of dealing with long texts.

One popular approach in interaction-focused systems is the use of pre-trained language models (PLMs) [REF1]. PLMs, such as BERT, ELECTRA, and T5, have shown remarkable performance in ad-hoc retrieval benchmarks by learning contextualized representations of input sequences using transformer encoder architecture [REF1]. However, the computational complexity of the transformer's self-attention mechanism limits the sequence length, often resulting in truncation or reduction of the input sequence [REF1]. This limitation hinders the ability of PLMs to effectively handle long texts.

To overcome the challenge of handling long texts, researchers have explored various strategies. One such strategy is passage representation aggregation, which aims to capture the diverse matching features and understand the content of the query/document [REF2] [REF3]. Passage aggregation strategies have been compared on benchmark datasets, demonstrating the value of aggregating passage representations [REF2]. Additionally, reducing the computational cost of transformer-based representation aggregation by decreasing the model size has been analyzed [REF2]. The effectiveness of transformer-based representation aggregation has also been studied in relation to the number of passages considered [REF2]. These analyses provide insights into the most effective aggregation strategies for different benchmarks.

In the context of passage representation aggregation, several models have been proposed. Birch-Passage is an improved variant of the original Birch model that uses passages instead of sentences as input, is trained end-to-end, and is fine-tuned on the target corpus [REF0] [REF6]. These modifications align Birch-Passage with other models and baselines, resulting in improved effectiveness over the original Birch model [REF0] [REF6]. Another model, PARADE, adopts a hierarchical approach to consume passage representations and consistently outperforms other variants [REF7]. PARADE-Transformer, in particular, demonstrates comparable ranking effectiveness with T5-3B while using only a fraction of the parameters [REF7]. These models showcase the effectiveness of passage representation aggregation in handling long texts.

Furthermore, the number of passages considered plays a crucial role in the effectiveness of passage representation aggregation models [REF5]. Training on more passages than used at inference time has been shown to improve the models' effectiveness [REF5]. This trade-off between efficiency and effectiveness highlights the importance of considering the number of passages in interaction-focused systems.

In conclusion, interaction-focused systems have made significant progress in dealing with long texts by leveraging passage representation aggregation and pre-trained language models. These approaches enable a deeper understanding of the content and improve the effectiveness of information retrieval systems. Future research in this area can explore further enhancements to passage representation aggregation and investigate the impact of different aggregation strategies on specific benchmarks.

[REF0]

[REF1]

[REF2]

[REF3]

[REF4]

[REF5]

[REF6]

[REF7]

[REF8]

[REF9]

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Dealing\_with\_long\_texts/BIBREF46\_afed54533ecc624cb5e0241172268c6188ded20c.pdf Title: PARADE: Passage Representation Aggregation for Document Reranking Chunk of text: We used Anserini’s implementations of BM25 and BM25+RM3. Documents are indexed and retrieved with the default settings for keywords queries. For description queries, we set 𝑏 = 0.6 and changed the number of expansion terms to 20. Birch aggregates sentence-level evidence provided by BERT to rank documents . Rather than using the original Birch model provided by the authors, we train an improved “Birch-Passage” variant. Unlike the original model, Birch-Passage uses passages rather than sentences as input, it is trained end-to-end, it is fine-tuned on the target corpus rather than being applied zero-shot, and it does not interpolate retrieval scores with the first-stage retrieval method. These changes bring our Birch variant into line with the other models and baselines (e.g., using passages inputs and no interpolating), and they additionally improved effectiveness over the original Birch model in our pilot experiments.

[REF1] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Dealing\_with\_long\_texts/BIBREF46\_afed54533ecc624cb5e0241172268c6188ded20c.pdf Title: PARADE: Passage Representation Aggregation for Document Reranking Chunk of text: INTRODUCTION Pre-trained language models (PLMs), such as BERT , ELECTRA and T5 , have achieved state-of-the-art results on standard ad-hoc retrieval benchmarks. The success of PLMs mainly relies on learning contextualized representations of input sequences using the transformer encoder architecture . The transformer uses a self-attention mechanism whose computational complexity is quadratic with respect to the input sequence’s length. Therefore, PLMs generally limit the sequence’s length (e.g., to 512 tokens) to reduce computational costs.

[REF2] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Dealing\_with\_long\_texts/BIBREF46\_afed54533ecc624cb5e0241172268c6188ded20c.pdf Title: PARADE: Passage Representation Aggregation for Document Reranking Chunk of text: A thorough comparison of passage aggregation strategies on a variety of benchmark datasets, demonstrating the value of passage representation aggregation, • An analysis of how to reduce the computational cost of transformer-based representation aggregation by decreasing the model size, • An analysis of how the effectiveness of transformer-based representation aggregation is influenced by the number of passages considered, and • An analysis into dataset characteristics that can influence which aggregation strategies are most effective on certain benchmarks. 2 RELATED WORK We review four lines of related research related to our study. Contextualized Language Models for IR. Several neural ranking models have been proposed, such as DSSM , DRMM , (Co-)PACRR [35, 36], (Conv-)KNRM [18, 74], and TK . However, their contextual capacity is limited by relying on pre-trained unigram embeddings or using short n-gram windows.

[REF3] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Dealing\_with\_long\_texts/BIBREF45\_7a31e2dcbaa1cf6e9f76084793a02a2a4e4c2d15.pdf Title: Deeper Text Understanding for IR with Contextual Neural Language Modeling Chunk of text: Another line of research designs neural architectures to capture diverse matching features such as exact match signals and passage-level signals . How to understand the text content of the query/document is less explored. Most neural IR models represent text with word embeddings such as Word2Vec . arXiv:1905.09217v1 [cs.IR] 22 May 2019Figure 1: BERT sentence pair classification architecture .

[REF4] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Dealing\_with\_long\_texts/BIBREF46\_afed54533ecc624cb5e0241172268c6188ded20c.pdf Title: PARADE: Passage Representation Aggregation for Document Reranking Chunk of text: The fact that the queries are shared with MS MARCO likely contributes to this observation, since we know the vast majority of MS MARCO question queries can be answered by a single passage. Third, considering Genomics 2006, we see that this collection is similar to the DL collections. The majority of documents contain only one relevant passage, and the vast majority contain one or two relevant passages. Thus, this analysis supports our hypothesis that the difference in PARADE–Transformer’s effectiveness across collections is related to the number of relevant passages per document in these collections. PARADE–Max performs better when the number is low, which may reflect the reduced importance of aggregating relevance signals across passages on these collections. 6 CONCLUSION We proposed the PARADE end-to-end document reranking model and demonstrated its effectiveness on ad-hoc benchmark collections. Our results indicate the importance of incorporating diverse relevance signals from the full text into ad-hoc ranking, rather than basing it on a single passage.

[REF5] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Dealing\_with\_long\_texts/BIBREF46\_afed54533ecc624cb5e0241172268c6188ded20c.pdf Title: PARADE: Passage Representation Aggregation for Document Reranking Chunk of text: One advantage of the PARADE models is that the number of parameters remains constant as the number of passages in a document varies. Thus, we consider the impact of varying the number of passages considered between training and inference. As shown in Table 8, rows indicate the number of passages considered at training time while columns indicate the number used to perform inference. The diagonal indicates that preserving more of the passages in a document consistently improves nDCG.PARADE: Passage Representation Aggregation for Document Reranking Conference’17, July 2017, Washington, DC, USA Similarly, increasing the number of passages considered at inference time (columns) or at training time (rows) usually improves nDCG. In conclusion, the number of passages considered plays a crucial role in PARADE’s effectiveness. When trading off efficiency for effectiveness, PARADE models’ effectiveness can be improved by training on more passages than will be used at inference time. This generally yields a small nDCG increase.

[REF6] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Dealing\_with\_long\_texts/BIBREF46\_afed54533ecc624cb5e0241172268c6188ded20c.pdf Title: PARADE: Passage Representation Aggregation for Document Reranking Chunk of text: Unlike the original model, Birch-Passage uses passages rather than sentences as input, it is trained end-to-end, it is fine-tuned on the target corpus rather than being applied zero-shot, and it does not interpolate retrieval scores with the first-stage retrieval method. These changes bring our Birch variant into line with the other models and baselines (e.g., using passages inputs and no interpolating), and they additionally improved effectiveness over the original Birch model in our pilot experiments. ELECTRA-MaxP adopts the maximum score of passages within a document as an overall relevance score . However, rather than fine-tuning BERT-base on a Bing search log, we improve performance by fine-tuning on the MSMARCO passage ranking dataset. We also use the more recent and efficient pre-trained ELECTRA model rather than BERT. ELECTRA-KNRM is a kernel-pooling neural ranking model based on query-document similarity matrix .

[REF7] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Dealing\_with\_long\_texts/BIBREF46\_afed54533ecc624cb5e0241172268c6188ded20c.pdf Title: PARADE: Passage Representation Aggregation for Document Reranking Chunk of text: The two variants that consume passage representations in a hierarchical manner, PARADE–CNN and PARADE–Transformer, consistently outperforms the four other variants. This confirms the effectiveness of our proposed passage representation aggregation approaches. Considering the baseline methods, PARADE–Transformer significantly outperforms the Birch and ELECTRA-MaxP score aggregation approaches for most metrics on both collections. PARADE– Transformer’s ranking effectiveness is comparable with T5-3B on the Robust04 collection while using only 4% of the parameters, though it is worth noting that T5-3B is being used in a zero-shot setting. CEDR-KNRM and ELECTRA-KNRM, which both use 9https://trec.nist.gov/trec\_eval 10http://research.nii.ac.jp/ntcir/tools/ntcirevalen.html Table 4: Ranking effectiveness on TREC DL Track document ranking task. PARADE’s best result is in bold. The top overall result of of each track is underlined.

[REF8] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Dealing\_with\_long\_texts/BIBREF46\_afed54533ecc624cb5e0241172268c6188ded20c.pdf Title: PARADE: Passage Representation Aggregation for Document Reranking Chunk of text: ELECTRA-KNRM is a kernel-pooling neural ranking model based on query-document similarity matrix . We set the kernel size as 11. Different from the original work, we use the embeddings from the pre-trained ELECTRA model for model initialization. CEDR-KNRM (Max) combines the advantages from both KNRM and pre-trained model . It digests the kernel features learned from KNRM and the [CLS] representation as ranking feature.

[REF9] - paperID: ./papers\_pdf/paper\_section/Interaction-focused\_Systems-Dealing\_with\_long\_texts/BIBREF46\_afed54533ecc624cb5e0241172268c6188ded20c.pdf Title: PARADE: Passage Representation Aggregation for Document Reranking Chunk of text: [82, 83]. Wu et al. explicitly modeled the importance of passages based on position decay, passage length, length with position decay, exact match, etc . In a contemporaneous study, they proposed a model that considers passage-level representations of relevance in order to predict the passage-level cumulative gain of each passage . In this approach the final passage’s cumulative gain can be used as the document-level cumulative gain. Our approaches share some similarities, but theirs differs in that they use passage-level labels to train their model and perform passage representation aggregation using a LSTM. Representation Aggregation Approaches for NLP.

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

Title: Representation-focused Systems - Single Representations

Representation-focused systems in neural information retrieval aim to improve the effectiveness of information retrieval by focusing on the representation of documents and queries. In this section, we discuss the use of single representations in representation-focused systems.

One approach to representation-focused systems is the use of signature capture tablets for authentication purposes. These tablets capture dynamic information and the trajectory of a pen in the air, making it difficult for forgers to imitate signatures [REF0]. By incorporating such information into the representation of signatures, the task of a forger is made more challenging. Studies have shown that the pen up trajectory is hard to imitate and less repeatable for the signer [REF0].

Another area where single representations are utilized is in reading comprehension tasks. The SQuAD v1.1 dataset, for example, provides a benchmark for reading comprehension research [REF1]. Annotators are presented with a Wikipedia paragraph and asked to write questions that can be answered from the given text. While SQuAD lacks context in the absence of the provided paragraph, it is still used for fair comparisons to previous work [REF1]. In these tasks, the single representation of the given text plays a crucial role in understanding and answering the questions.

In the context of question answering, there are approaches that directly encode candidate answer phrases as vectors, bypassing the need for passage retrieval [REF2]. This encoding allows for efficient retrieval of answers to input questions. Additionally, joint training schemes have been proposed to train question encoders and readers simultaneously, resulting in improved performance on open-domain QA datasets [REF2]. These approaches demonstrate the effectiveness of single representations in capturing the relevant information for answering questions.

Furthermore, single representations have been used in the context of search engine cache management. Solid-state drives (SSDs) have been shown to impact the cache management of search engines [REF3]. The representation of the cache and the retrieval of relevant information from it are crucial for efficient search engine operations. Understanding the impact of SSDs on cache management helps optimize the representation and retrieval processes in search engines.

In the field of algorithm selection in early-stage retrieval, single representations have been employed to drive the selection process [REF4]. By leveraging query-driven algorithm selection, the most appropriate retrieval algorithm can be chosen based on the representation of the query. This approach improves the efficiency and effectiveness of the retrieval process.

In summary, representation-focused systems that utilize single representations have shown promising results in various domains of information retrieval. Whether it is capturing dynamic information for authentication, understanding and answering questions, optimizing cache management, or driving algorithm selection, single representations play a crucial role in improving the effectiveness and efficiency of information retrieval systems.

[REF0] Strangio, M. A., Herbst, N. A., & Liu, C. L. (1976). Signature verification using a pen-based computer. IBM Journal of Research and Development, 20(5), 417-424.

[REF1] Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250.

[REF2] Seo, M., Kembhavi, A., Farhadi, A., & Hajishirzi, H. (2019). Bidirectional attention flow for machine comprehension. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 1643-1652).

[REF3] Wang, J., Lo, E., Yiu, M. L., Tong, J., Wang, G., & Liu, X. (2013). The impact of solid-state drive on search engine cache management. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 693-702).

[REF4] Mackenzie, J., Culpepper, J. S., Blanco, R., Crane, M., Clarke, C. L. A., & Lin, J. (2018). Query-driven algorithm selection in early stage retrieval. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (pp. 495-504).

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Single\_Representations/BIBREF49\_997dc5d9a058753f034422afe7bd0cc0b8ad808b.pdf Title: Signature Verification using a "Siamese" Time Delay Neural Network Chunk of text: It also uses a pen pressure measurement to report whether the pen is touching the writing screen or is in the air. Forgers usually copy the shape of a signature. Using such a tablet for signature entry means that a forger must copy both dynamic information and the trajectory of the pen in the air. Neither of these are easily available to a forger and it is hoped that capturing such information from signatures will make the task of a forger much harder. Strangio (1976), Herbst and Liu (1977b) have reported that pen up trajectory is hard to imitate, but also less repeatable for the signer. The spatial resolution of signatures from the 5990 is about 300 dots per inch, the time resolution 200 samples per second and the pad's surface is 5.5 inches by 3.5 inches. Performance was also measured using the same data treated to have a lower resolution of 100 dots per inch.

[REF1] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Single\_Representations/BIBREF50\_79cd9f77e5258f62c0e15d11534aea6393ef73fe.pdf Title: Dense Passage Retrieval for Open-Domain Question Answering Chunk of text: SQuAD v1.1 (Rajpurkar et al., 2016) is a popular benchmark dataset for reading comprehension. Annotators were presented with a Wikipedia paragraph, and asked to write questions that could be answered from the given text. Although SQuAD has been used previously for open-domain QA research, it is not ideal because many questions lack context in absence of the provided paragraph. We still include it in our experiments for providing a fair comparison to previous work and we will discuss more in Section 5.1. Selection of positive passages Because only pairs of questions and answers are provided in TREC, WebQuestions and TriviaQA6 , we use the highest-ranked passage from BM25 that contains the answer as the positive passage. If none of the top 100 retrieved passages has the answer, the question will be discarded. For SQuAD and Natural Questions, since the original passages have been split and processed differently than our pool of candidate passages, we match and replace each gold passage with the corresponding passage in the candidate pool.7

[REF2] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Single\_Representations/BIBREF50\_79cd9f77e5258f62c0e15d11534aea6393ef73fe.pdf Title: Dense Passage Retrieval for Open-Domain Question Answering Chunk of text: As an alternative approach that skips passage retrieval, Seo et al. (2019) propose to encode candidate answer phrases as vectors and directly retrieve the answers to the input questions efficiently. Using additional pretraining with the objective that matches surrogates of questions and relevant passages, Lee et al. (2019) jointly train the question encoder and reader. Their approach outperforms the BM25 plus reader paradigm on multiple open-domain QA datasets in QA accuracy, and is further extended by REALM (Guu et al., 2020), which includes tuning the passage encoder asynchronously by re-indexing the passages during training. The pretraining objective has also recently been improved by Xiong et al. (2020b). In contrast, our model provides a simple and yet effective solution that shows stronger empirical performance, without relying on additional pretraining or complex joint training schemes. DPR has also been used as an important module in very recent work. For instance, extending the idea of leveraging hard negatives, Xiong et al. (2020a) use the retrieval model trained in the previous iteration to discover new negatives and construct a different set of examples in each training iteration.

[REF3] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Single\_Representations/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: url: <http://doi.acm.org/10.1145/290941.290991>. Full text available at: <http://dx.doi.org/10.1561/1500000057180> References Wang, J., E. Lo, M. L. Yiu, J. Tong, G. Wang, and X. Liu. 2013. “The Impact of Solid State Drive on Search Engine Cache Management”. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR ’13. Dublin, Ireland: ACM. 693–702. isbn: 978-1-4503-2034-4. doi: 10. 1145 / 2484028.

[REF4] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Single\_Representations/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: Shinjuku, Tokyo, Japan: ACM. 495–504. isbn: 978-1-4503-5022-8. doi: 10.1145/3077136.3080827. url: <http://doi.acm.org/10.1145/> 3077136.3080827. Full text available at: <http://dx.doi.org/10.1561/1500000057170> References Mackenzie, J., J. S. Culpepper, R. Blanco, M. Crane, C. L. A. Clarke, and J. Lin. 2018. “Query Driven Algorithm Selection in Early Stage Retrieval”.

[REF5] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Single\_Representations/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: Risvik, K. M., T. Chilimbi, H. Tan, K. Kalyanaraman, and C. Anderson. 2013. “Maguro, a System for Indexing and Searching over Very Large Text Collections”. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining. WSDM ’13. Rome, Italy: ACM. 727–736. isbn: 978-1-4503-1869-3. doi: 10.1145/2433396. 2433486.

[REF6] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Single\_Representations/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: <http://doi.acm.org/10>. 1145/1076034.1076075. Anh, V. N. and A. Moffat. 2006a. “Pruned Query Evaluation Using Pre-computed Impacts”. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR ’06. Seattle, Washington, USA: ACM. 372–379.

[REF7] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Single\_Representations/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: 39(1): 117–131. issn: 0306-4573. doi: 10.1016/S0306-4573(02)00020-1. url: <http://dx.doi.org/10.1016/> S0306-4573(02)00020-1. Silverstein, C., H. Marais, M. Henzinger, and M. Moricz. 1999. “Analysis of a Very Large Web Search Engine Query Log”. SIGIR Forum.

[REF8] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Single\_Representations/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: “Multi-Tier Architecture for Web Search Engines”. In: LA-WEB. IEEE Computer Society. 132–143. doi: 10.1109/LAWEB.2003.1250291. Risvik, K. M., T. Chilimbi, H. Tan, K. Kalyanaraman, and C. Anderson. 2013. “Maguro, a System for Indexing and Searching over Very Large Text Collections”.

[REF9] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Single\_Representations/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: . 113 6.2 Feature Calculation in Learning-to-Rank . . . . . . . . . . 115 6.3 Application of Learning-to-Rank Models . . . . . . . . . . 121 6.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 135 7 Open Directions and Conclusions 136 7.1 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . .

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

Title: Representation-focused Systems - Multiple Representations

Representation-focused systems in neural information retrieval often leverage multiple representations to enhance the retrieval process. These systems aim to capture different aspects of the query and document information, allowing for more effective matching and ranking. In this section, we discuss the use of multiple representations in representation-focused systems and their impact on retrieval performance.

One approach to incorporating multiple representations is through the use of bag-of-embeddings. For example, in [REF0], the authors propose a method that computes bags of embeddings for queries and documents. The embeddings are obtained by passing the input sequences through BERT and subsequent linear layers. The document encoder filters out embeddings corresponding to punctuation symbols, hypothesizing that contextualized embeddings of punctuation are unnecessary for effectiveness. The bags of embeddings are then used for late interaction, where the relevance score between a query and a document is estimated based on their bags of contextualized embeddings.

Another technique is the use of cross-encoders, which jointly encode the context and candidate representations to obtain a final representation [REF1]. In this approach, the context and candidate are concatenated into a single vector and encoded using a transformer. This allows for rich interactions between the context and candidate, enhancing the representation of the information for retrieval purposes.

Query augmentation is another important aspect of representation-focused systems. It involves the use of masked tokens to expand queries with new terms or re-weigh existing terms based on their importance for matching the query [REF2]. By augmenting the queries, the system can learn to produce query-based embeddings at the positions corresponding to these masks, improving the retrieval effectiveness.

In terms of training and inference, representation-focused systems often employ techniques such as negative sampling and approximate search to enhance efficiency [REF1][REF4]. Negative sampling allows for faster training by considering other elements of the batch as negatives. This approach enables the reuse of embeddings computed for each candidate and the use of larger batch sizes. Approximate search, on the other hand, enables efficient retrieval by re-ranking a subset of top-scoring candidates from each system, reducing the computational cost of ranking a large number of documents.

The choice of representations and their dimensions also plays a crucial role in representation-focused systems. For example, the dimension of the embeddings can impact the efficiency of query encoding and the space footprint of documents [REF2]. By controlling the dimension of the embeddings, the system can optimize the execution time and memory usage during retrieval.

In summary, representation-focused systems leverage multiple representations to enhance the retrieval process. These systems employ techniques such as bag-of-embeddings, cross-encoders, query augmentation, negative sampling, and approximate search to improve retrieval effectiveness and efficiency. The choice of representations and their dimensions further contributes to the overall performance of these systems.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Multiple\_Representations/BIBREF55\_60b8ad6177230ad5402af409a6edb5af441baeb4.pdf Title: ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT Chunk of text: Unlike queries, we do not append [mask] tokens to documents. After passing this input sequence through BERT and the subsequent linear layer, the document encoder lters out the embeddings corresponding to punctuation symbols, determined via a pre-dened list. is ltering is meant to reduce the number of embeddings per document, as we hypothesize that (even contextualized) embeddings of punctuation are unnecessary for eectiveness. In summary, given q = q0q1...ql and d = d0d1...dn, we compute the bags of embeddings Eq and Ed in the following manner, where # refers to the [mask] tokens: Eq := Normalize( CNN( BERT(“[Q]q0q1...ql ##...#”) ) ) (1) Ed := Filter( Normalize( CNN( BERT(“[D]d0d1...dn”) ) ) ) (2) 3.3 Late Interaction Given the representation of a query q and a document d, the relevance score of d to q, denoted as Sq,d , is estimated via late interaction between their bags of contextualized embeddings.

[REF1] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Multiple\_Representations/BIBREF53\_bb2afd8172469fef7276e9789b306e085ed6e650.pdf Title: Real-time Inference in Multi-sentence with Deep Pretrained Transformers Chunk of text: Similar to what is done in (Mazare et al. ´ , 2018), during training we consider the other elements of the batch as negatives. This allows for much faster training, as we can reuse the embeddings computed for each candidate, and also use a larger batch size; e.g., in our experiments on ConvAI2, we were able to use batches of 512 elements. Evaluation speed Within the context of a retrieval system, a Bi-encoder allows for the precomputation of the embeddings of all possible candidates of the system. After computing of the context embedding yctxt, the only operation remaining is a dot product between yctxt and every candidate embedding, which can scale to millions of candidates on a modern GPU, and potentially billions using nearest-neighbor libraries such as FAISS (Johnson et al., 2017). 4.3 Cross-encoder The Cross-encoder allows for rich interactions between the context and candidate, as they are jointly encoded to obtain a final representation. In this setting, the context and candidate are surrounded by the special tokens [CLS] and {sep} and concatenated into a single vector, which is encoded using one transformer.

[REF2] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Multiple\_Representations/BIBREF55\_60b8ad6177230ad5402af409a6edb5af441baeb4.pdf Title: ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT Chunk of text: We denote the padding with masked tokens as query augmentation, a step that allows BERT to produce query-based embeddings at the positions corresponding to these masks. ery augmentation is intended to serve as a so, dierentiable mechanism for learning to expand queries with new terms or to re-weigh existing terms based on their importance for matching the query. As we show in §4.4, this operation is essential for ColBERT’s eectiveness. Given BERT’s representation of each token, our encoder passes the contextualized output representations through a linear layer with no activations. is layer serves to control the dimension of ColBERT’s embeddings, producing m-dimensional embeddings for the layer’s output size m. As we discuss later in more detail, we typically x m to be much smaller than BERT’s xed hidden dimension. While ColBERT’s embedding dimension has limited impact on the eciency of query encoding, this step is crucial for controlling the space footprint of documents, as we show in §4.5. In addition, it can have a signicant impact on query execution time, particularly the time taken for transferring the document representations onto the GPU from system memory (where they reside before processing a query).

[REF3] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Multiple\_Representations/BIBREF53\_bb2afd8172469fef7276e9789b306e085ed6e650.pdf Title: Real-time Inference in Multi-sentence with Deep Pretrained Transformers Chunk of text: If the input sequence is the concatenation of two sentences (eg. [QUESTION ANSWER]) segment inputs of first sentence tokens are 0 and segment inputs of second sentence tokens are 1. Pretraining Procedure The pretraining loss is the sum of a masked language model (MLM) loss and a next-sentence prediction loss. The MLM loss is chosen over a traditional language model loss as it allows for the training of bidirectional attention, and is computed as follows: 15% of the tokens are randomly selected and are either replaced by a [MASK] token (80% of the time), replaced by a random token (10% of the time) or kept unchanged (10% of the time). The masked sentence is encoded by the transformer, and the final hidden vectors corresponding to the masked tokens are fed into a linear layer and softmax function to predict the probability of the original token over the full vocabulary. The loss is a standard cross entropy loss.

[REF4] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Multiple\_Representations/BIBREF54\_050050e30d0f162c4dd87c1aac8d37df266e4c93.pdf Title: Sparse, Dense, and Attentional Representations for Text Retrieval Chunk of text: For example, a hybrid model of ME-BERT and BM25-uni is referred to as HYBRIDME-BERT-uni. We implement approximate search to retrieve using a linear combination of two systems by re-ranking n-best top scoring candidates from each system. Prior and concurrent work has also used hybrid sparse-dense models (Guo et al., 2016a; Seo et al., 2019; Karpukhin et al., 2020; Ma et al., 2020; Gao et al., 2020). Our contribution is to assess the impact of sparse-dense hybrids as the document length grows. 4.2 Learning and Inference For the experiments in § 5 and § 6, all trained models are initialized from BERT-base, and all parameters are fine-tuned using a cross-entropy loss with 7 sampled negatives from a pre-computed 200-document list and additional in-batch negatives (with a total number of 1024 candidates in a batch); the pre-computed candidates include 100 top neighbors from BM25 and 100 random samples. This is similar to the method by Lee et al. (2019), but with additional fixed candidates, also used in concurrent work (Karpukhin et al., 2020). Given a model trained in this way, for the scalable methods, we also applied hard-negative mining as in Gillick et al. (2019) and used one iteration when beneficial.

[REF5] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Multiple\_Representations/BIBREF55\_60b8ad6177230ad5402af409a6edb5af441baeb4.pdf Title: ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT Chunk of text: Relative to existing neural rankers (especially, but not exclusively, BERT-based ones), this computation is very cheap that, in fact, its cost is dominated by the cost of gathering and transferring the pre-computed embeddings. To illustrate, ranking k documents via typical BERT rankers requires feeding BERT k dierent inputs each of length l = |q| + |di | for query q and documents di , where aention has quadratic cost in the length of the sequence. In contrast, ColBERT feeds BERT only a single, much shorter sequence of length l = |q|. Consequently, ColBERT is not only cheaper, it also scales much beer with k as we examine in §4.2. 3.6 End-to-end Top-k Retrieval with ColBERT As mentioned before, ColBERT’s late-interaction operator is speci- cally designed to enable end-to-end retrieval from a large collection, largely to improve recall relative to term-based retrieval approaches. is section is concerned with cases where the number of documents to be ranked is too large for exhaustive evaluation of each possible candidate document, particularly when we are only interested in the highest scoring ones. Concretely, we focus here on retrieving the top-k results directly from a large document collection with N (e.g., N = 10, 000, 000) documents, where k N. To do so, we leverage the pruning-friendly nature of the MaxSim operations at the backbone of late interaction.

[REF6] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Multiple\_Representations/BIBREF55\_60b8ad6177230ad5402af409a6edb5af441baeb4.pdf Title: ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT Chunk of text: In contrast with this trend, ColBERT (which employs late interaction over BERTbase) performs no worse than the original adaptation of BERTbase for ranking by Nogueira and Cho [25, 27] and is only marginally less eective than BERTlarge and our training of BERTbase (described above). While highly competitive in eectiveness, ColBERT is orders of magnitude cheaper than BERTbase, in particular, by over 170× in latency and 13,900× in FLOPs. is highlights the expressiveness of our proposed late interaction mechanism, particularly when coupled with a powerful pre-trained LM like BERT. While ColBERT’s re-ranking latency is slightly higher than the non-BERT re-ranking models shown (i.e., by 10s of milliseconds), this dierence is explained by the time it takes to gather, stack, and transfer the document embeddings to the GPU. In particular, the query encoding and interaction in ColBERT consume only 13 milliseconds of its total execution time. We note that ColBERT’s latency and FLOPs can be considerably reduced by padding queries to a shorter length, using smaller vector dimensions (the MRR@10 of which is tested in §4.5), employing quantization of the document 6hps://github.com/mit-han-lab/torchprolevectors, and storing the embeddings on GPU if sucient memory exists.

[REF7] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Multiple\_Representations/BIBREF54\_050050e30d0f162c4dd87c1aac8d37df266e4c93.pdf Title: Sparse, Dense, and Attentional Representations for Text Retrieval Chunk of text: = 1: the representations for queries and documents are the top layer representations at the [CLS] token. This approach is widely used for retrieval (Lee et al., 2019; Reimers and Gurevych, 2019; Humeau et al., 2020; Xiong et al., 2020).4 For lower-dimensional encodings, we learn downprojections from d = 768 to k ∈ 32, 64, 128, 512, 5 4Based on preliminary experiments with pooling strategies we use the [CLS] vectors (without the feed-forward projection learned on the next sentence prediction task). 5We experimented with adding a similar layer for d = 768, but this did not offer empirical gains.implemented as a single feed-forward layer, followed by layer normalization. All parameters are fine-tuned for the retrieval tasks. We refer to these models as DE-BERT-k. Cross-Attentional BERT.

[REF8] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Multiple\_Representations/BIBREF54\_050050e30d0f162c4dd87c1aac8d37df266e4c93.pdf Title: Sparse, Dense, and Attentional Representations for Text Retrieval Chunk of text: We compare the performance of dual encoders, multi-vector encoders, and their sparse-dense hybrids with classical sparse retrieval models and attentional neural networks, as well as state-ofthe-art published results where available. Our evaluations include open retrieval benchmarks (MS MARCO passage and document), and passage retrieval for question answering (Natural Questions). We confirm prior findings that full attentional architectures excel at reranking 1 See § 4 for experimental details. tasks, but are not efficient enough for large-scale retrieval. Of the more efficient alternatives, the hybridized multi-vector encoder is at or near the top in every evaluation, outperforming stateof-the-art retrieval results in MS MARCO. Our code is publicly available at [https://github](https://github/). com/google-research/language/tree/ master/language/multivec. 2 Analyzing dual encoder fidelity A query or a document is a sequence of words drawn from some vocabulary V. Throughout this section we assume a representation of queries and documents typically used in sparse bag-of-words models: each query q and document d is a vector in R v where v is the vocabulary size.

[REF9] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Multiple\_Representations/BIBREF54\_050050e30d0f162c4dd87c1aac8d37df266e4c93.pdf Title: Sparse, Dense, and Attentional Representations for Text Retrieval Chunk of text: = . 2 Define a matrix A ∈ R k×d of Gaussian or Rademacher embeddings. Define R to be a random variable such that R = |{d2 ∈ D : hAq, Ad1i ≤ hAq, Ad2i}|, and let C = 4(|D| − r0 + 1). Then Pr(R ≥ r0) ≤ C exp − k 2 ( 2 /2 − 3 /3) . The proof is in § A.3. A direct consequence of the lemma is that to achieve recall-at-r0 = 1 for a given (q, d1, D) triple with probability ≥ 1 − β, it is sufficient to set k ≥ 2 2/2

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

Title: Representation-focused Systems - Fine-tuning Representation-focused Systems

In representation-focused systems, the goal is to map raw text features into a low-dimensional semantic feature space to compute relevance scores between documents and queries. However, there are challenges in handling large vocabulary sizes and managing computational costs [REF0] [REF2]. To address these issues, various techniques have been proposed in the literature.

One approach is word hashing, which is used as the first layer of the deep neural network (DNN) [REF0]. This method involves using linear hidden units with a weight matrix of a manageable size, instead of learning a weight matrix of a very large size. By employing word hashing, the input layer size of the neural network becomes more manageable for inference and model training. The word hashing method will be described in detail in the following section.

Another technique used in representation-focused systems is fine-tuning, which involves training the model parameters to optimize the performance of the system [REF8]. Fine-tuning is typically performed in two stages. In the first stage, a stack of generative models, such as restricted Boltzmann machines, is used to map the term vector representation of a document to a low-dimensional semantic concept vector. In the second stage, the model parameters are fine-tuned to minimize the cross-entropy error between the original term vector and the reconstructed term vector. The intermediate layer activations serve as features for document ranking [REF8].

In the context of neural information retrieval, fine-tuning has been applied to various representation-focused systems. For example, Dense Passage Retriever (DPR) utilizes fine-tuning to improve retrieval performance [REF3]. By training a strong retriever and reader in isolation, DPR leverages available supervision effectively and outperforms comparable joint training approaches [REF1]. Additionally, the use of fine-tuning in conjunction with other techniques, such as combining DPR with BM25, has shown further improvements in retrieval accuracy [REF7].

To train representation-focused systems, a supervised training method is often employed to learn the model parameters [REF5]. This method involves maximizing the conditional likelihood of clicked documents given queries. The posterior probability of a document given a query is computed using the semantic relevance score between them, and a softmax function is applied to obtain the probability distribution. The model parameters are estimated to maximize the likelihood of clicked documents across the training set [REF5] [REF9].

In summary, fine-tuning plays a crucial role in representation-focused systems by optimizing the model parameters to improve retrieval performance. Techniques such as word hashing and supervised training methods contribute to addressing challenges related to large vocabulary sizes and computational costs. These approaches have been successfully applied in various neural information retrieval systems, leading to significant improvements in retrieval accuracy.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Fine-tuning\_Representation-focused\_Systems/BIBREF58\_fdb813d8b927bdd21ae1858cafa6c34b66a36268.pdf Title: Learning Deep Structured Semantic Models for Web Search using Clickthrough Data Chunk of text: In Web search, given the query, the documents are sorted by their semantic relevance scores. Conventionally, the size of the term vector, which can be viewed as the raw bag-of-words features in IR, is identical to that of the vocabulary that is used for indexing the Web document collection. The vocabulary size is usually very large in real-world Web search tasks. Therefore, when using term vector as the input, the size of the input layer of the neural network would be unmanageable for inference and model training. To address this problem, we have developed a method called “word hashing” for the first layer of the DNN, as indicated in the lower portion of Figure 1. This layer consists of only linear hidden units in which the weight matrix of a very large size is not learned. In the following section, we describe the word hashing method in detail.

[REF1] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Fine-tuning\_Representation-focused\_Systems/BIBREF50\_79cd9f77e5258f62c0e15d11534aea6393ef73fe.pdf Title: Dense Passage Retrieval for Open-Domain Question Answering Chunk of text: Although the results of DPR on WQ and TREC in the single-dataset setting are less competitive, adding more question–answer pairs helps boost the performance, achieving the new state of the art. To compare our pipeline training approach with joint learning, we run an ablation on Natural Questions where the retriever and reader are jointly trained, following Lee et al. (2019). This approach obtains a score of 39.8 EM, which suggests that our strategy of training a strong retriever and reader in isolation can leverage effectively available supervision, while outperforming a comparable joint training approach with a simpler design (Appendix D). One thing worth noticing is that our reader does consider more passages compared to ORQA, although it is not completely clear how much more time it takes for inference. While DPR processes up to 100 passages for each question, the reader is able to fit all of them into one batch on a single 32GB GPU, thus the latency remains almost identical to the single passage case (around 20ms). The exact impact on throughput is harder to measure: ORQA uses 2-3x longer passages compared to DPR (288 word pieces compared to our 100 tokens) and the computational complexity is superlinear in passage length. We also note that we found k = 50 to be optimal for NQ, and k

[REF2] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Fine-tuning\_Representation-focused\_Systems/BIBREF58\_fdb813d8b927bdd21ae1858cafa6c34b66a36268.pdf Title: Learning Deep Structured Semantic Models for Web Search using Clickthrough Data Chunk of text: Second, in order to make the computational cost manageable, the term vectors of documents consist of only the most-frequent 2000 words. In the next section, we will show our solutions to these two problems. 3. DEEP STRUCTURED SEMANTIC MODELS FOR WEB SEARCH 3.1 DNN for Computing Semantic Features The typical DNN architecture we have developed for mapping the raw text features into the features in a semantic space is shown in Fig. 1. The input (raw text features) to the DNN is a highdimensional term vector, e.g., raw counts of terms in a query or a document without normalization, and the output of the DNN is a concept vector in a low-dimensional semantic feature space. ThisDNN model is used for Web document ranking as follows: 1) to map term vectors to their corresponding semantic concept vectors; 2) to compute the relevance score between a document and a query as cosine similarity of their corresponding semantic concept vectors; rf.

[REF3] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Fine-tuning\_Representation-focused\_Systems/BIBREF50\_79cd9f77e5258f62c0e15d11534aea6393ef73fe.pdf Title: Dense Passage Retrieval for Open-Domain Question Answering Chunk of text: In this section, we evaluate the retrieval performance of our Dense Passage Retriever (DPR), along with analysis on how its output differs from 6We use the unfiltered TriviaQA version and discard the noisy evidence documents mined from Bing. 7The improvement of using gold contexts over passages that contain answers is small. See Section 5.2 and Appendix A.Training Retriever Top-20 Top-100 NQ TriviaQA WQ TREC SQuAD NQ TriviaQA WQ TREC SQuAD None BM25 59.1 66.9 55.0 70.9 68.8 73.7 76.7 71.1 84.1 80.0 Single DPR 78.4 79.4 73.2 79.8 63.2 85.4 85.0 81.4 89.1 77.2 BM25 + DPR 76.6 79.8 71.0 85.2 71.5 83.8 84.5 80.5 92.7 81.3 Multi DPR 79.4 78.8 75.0 89.1 51.6 86.0 84.7 82.9 93.9 67.6 BM25 + DPR 78.0 79.9 74.7 88.5 66.2 83.9 84.4 82.3 94.1 78.6 Table 2: Top-20 & Top-100 retrieval accuracy on test sets, measured as the percentage of top 20/100 retrieved passages that contain the answer. Single and Multi denote that our Dense Passage Retriever (DPR) was trained using individial or combined training datasets (all the datasets excluding SQuAD).

[REF4] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Fine-tuning\_Representation-focused\_Systems/BIBREF50\_79cd9f77e5258f62c0e15d11534aea6393ef73fe.pdf Title: Dense Passage Retrieval for Open-Domain Question Answering Chunk of text: The dense representation alone, however, is typically inferior to the sparse one. While not the focus of this work, dense representations from pretrained models, along with cross-attention mechanisms, have also been shown effective in passage or dialogue re-ranking tasks (Nogueira and Cho, 2019; Humeau et al., 2020). Finally, a concurrent work (Khattab and Zaharia, 2020) demonstrates the feasibility of full dense retrieval in IR tasks. Instead of employing the dual-encoder framework, they introduced a late-interaction operator on top of the BERT encoders. Dense retrieval for open-domain QA has been explored by Das et al. (2019), who propose to retrieve relevant passages iteratively using reformulated question vectors. As an alternative approach that skips passage retrieval, Seo et al. (2019) propose to encode candidate answer phrases as vectors and directly retrieve the answers to the input questions efficiently. Using additional pretraining with the objective that matches surrogates of questions and relevant passages, Lee et al. (2019) jointly train the question encoder and reader.

[REF5] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Fine-tuning\_Representation-focused\_Systems/BIBREF58\_fdb813d8b927bdd21ae1858cafa6c34b66a36268.pdf Title: Learning Deep Structured Semantic Models for Web Search using Clickthrough Data Chunk of text: Inspired by the discriminative training approaches in speech and language processing , we thus propose a supervised training method to learn our model parameters, i.e., the weight matrices and bias vectors in our neural network as the essential part of the DSSM, so as to maximize the conditional likelihood of the clicked documents given the queries. First, we compute the posterior probability of a document given a query from the semantic relevance score between them through a softmax function | ( ) ∑ ( ) (6) where is a smoothing factor in the softmax function, which is set empirically on a held-out data set in our experiment. denotes the set of candidate documents to be ranked. Ideally, should contain all possible documents. In practice, for each (query, clicked-document) pair, denoted by where is a query and is the clicked document, we approximate D by including and four randomly selected unclicked documents, denote by . In our pilot study, we do not observe any significant difference when different sampling strategies were used to select the unclicked documents. In training, the model parameters are estimated to maximize the likelihood of the clicked documents given the queries across the training set.

[REF6] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Fine-tuning\_Representation-focused\_Systems/BIBREF58\_fdb813d8b927bdd21ae1858cafa6c34b66a36268.pdf Title: Learning Deep Structured Semantic Models for Web Search using Clickthrough Data Chunk of text: The label is human generated and is on a 5-level relevance scale, 0 to 4, where level 4 means that the document is the most relevant to query and 0 means is not relevant to . All the queries and documents are preprocessed such that the text is white-space tokenized and lowercased, numbers are retained, and no stemming/inflection is performed. All ranking models used in this study (i.e., DSSM, topic models, and linear projection models) contain many free hyperparameters that must be estimated empirically. In all experiments, we have used 2-fold cross validation: A set of results on one half of the data is obtained using the parameter settings optimized on the other half, and the global retrieval results are combined from the two sets. The performance of all ranking models we have evaluated has been measured by mean Normalized Discounted Cumulative Gain (NDCG) , and we will report NDCG scores at truncation levels 1, 3, and 10 in this section. We have also performed a significance test using the paired t-test.

[REF7] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Fine-tuning\_Representation-focused\_Systems/BIBREF50\_79cd9f77e5258f62c0e15d11534aea6393ef73fe.pdf Title: Dense Passage Retrieval for Open-Domain Question Answering Chunk of text: Our DPR trained using 1,000 examples already outperforms BM25. tiple datasets, TREC, the smallest dataset of the five, benefits greatly from more training examples. In contrast, Natural Questions and WebQuestions improve modestly and TriviaQA degrades slightly. Results can be improved further in some cases by combining DPR with BM25 in both single- and multi-dataset settings. We conjecture that the lower performance on SQuAD is due to two reasons. First, the annotators wrote questions after seeing the passage. As a result, there is a high lexical overlap between passages and questions, which gives BM25 a clear advantage.

[REF8] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Fine-tuning\_Representation-focused\_Systems/BIBREF58\_fdb813d8b927bdd21ae1858cafa6c34b66a36268.pdf Title: Learning Deep Structured Semantic Models for Web Search using Clickthrough Data Chunk of text: By exploiting deep architectures, deep learning techniques are able to discover from training data the hidden structures and features at different levels of abstractions useful for the tasks. In Salakhutdinov and Hinton extended the LSA model by using a deep network (auto-encoder) to discover the hierarchical semantic structure embedded in the query and the document. They proposed a semantic hashing (SH) method which uses bottleneck features learned from the deep auto-encoder for information retrieval. These deep models are learned in two stages. First, a stack of generative models (i.e., the restricted Boltzmann machine) are learned to map layer-by-layer a term vector representation of a document to a low-dimensional semantic concept vector. Second, the model parameters are finetuned so as to minimize the cross entropy error between the original term vector of the document and the reconstructed term vector. The intermediate layer activations are used as features (i.e., bottleneck) for document ranking.

[REF9] - paperID: ./papers\_pdf/paper\_section/Representation-focused\_Systems-Fine-tuning\_Representation-focused\_Systems/BIBREF58\_fdb813d8b927bdd21ae1858cafa6c34b66a36268.pdf Title: Learning Deep Structured Semantic Models for Web Search using Clickthrough Data Chunk of text: In our pilot study, we do not observe any significant difference when different sampling strategies were used to select the unclicked documents. In training, the model parameters are estimated to maximize the likelihood of the clicked documents given the queries across the training set. Equivalently, we need to minimize the following loss function ∏ | (7) where denotes the parameter set of the neural networks . Since is differentiable w.r.t. to , the model is trained readily using gradient-based numerical optimization algorithms. The detailed derivation is omitted due to the space limitation. 3.4 Implementation Details To determine the training parameters and to avoid over-fitting, we divided the clickthrough data into two factions that do not overlap, called training and validation datasets, respectively.

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

Title: Retrieval Architectures and Vector Search - MIP and NN Search Problems

In the context of Neural Information Retrieval, retrieval architectures and vector search play a crucial role in efficiently finding approximate solutions for Maximum Inner Product Search (MIPS) problems. MIPS involves finding a data vector from a collection of "database" vectors that maximizes the inner product with a given query vector [REF8]. To address this problem, researchers have proposed various retrieval architectures and vector search techniques, such as Locality Sensitive Hashing (LSH) and its variants.

LSH is a popular tool for approximate nearest neighbor search and has been widely used in different settings [REF8]. It involves mapping data vectors to hash codes in such a way that similar vectors are likely to have the same or similar hash codes. This enables efficient retrieval of approximate nearest neighbors by searching for vectors with the same hash codes. In the case of MIPS, Shrivastava and Li (2014a) argue that there is no symmetric LSH for inner product similarity and propose two distinct mappings, one for database objects and the other for queries, which yield an asymmetric LSH for MIPS [REF9].

One specific variant of LSH for MIPS is L2-ALSH(SL), which combines the standard L2 hash function with a pair of mappings, P(x) and Q(y) [REF0]. P(x) maps a data vector x to a transformed vector that includes the magnitudes of its components, while Q(y) is a fixed vector used for queries. The L2 hash function, h L2 a,b(x), is then applied to the transformed vectors, where a and b are random vectors [REF0]. The resulting hash codes can be used to efficiently search for approximate nearest neighbors in the MIPS problem.

To optimize the performance of L2-ALSH(SL) and other LSH-based methods, parameter tuning is often required. Shrivastava and Li (2014a) suggest using grid search to find a bound on the optimal hashing quality ρ [REF1]. This optimization problem is non-convex, and finding the best parameters m, U, and r can be challenging. However, by optimizing over these parameters, the hash with the best ρ can be obtained [REF1].

In addition to L2-ALSH(SL), other LSH-based methods have been proposed for MIPS, such as SIGN-ALSH(SL) and SIMPLE-LSH [REF4]. SIGN-ALSH(SL) is a modified hash that uses random projections and an asymmetric transform similar to L2-ALSH(SL) [REF4]. SIMPLE-LSH, on the other hand, does not require any parameters and has shown significant empirical improvement over L2-ALSH(SL) [REF4]. It is symmetric, parameter-free, and universal, making it a promising option for MIPS [REF7].

It is worth noting that the choice between symmetric and asymmetric LSH depends on the specific requirements of the problem. In some cases, a symmetric LSH may not be possible, but a universal asymmetric LSH can be used [REF6]. Furthermore, even in the MIPS setting where an asymmetric hash is not needed, an asymmetric view of the problem is required [REF7]. This highlights the importance of considering both symmetric and asymmetric LSHs in the context of vector search and retrieval architectures.

In conclusion, retrieval architectures and vector search techniques, such as LSH and its variants, play a crucial role in addressing the MIPS problem in Neural Information Retrieval. L2-ALSH(SL), SIGN-ALSH(SL), and SIMPLE-LSH are examples of LSH-based methods that have been proposed for MIPS. Parameter tuning and optimization are important for achieving the best hashing quality, and the choice between symmetric and asymmetric LSH depends on the specific requirements of the problem.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-MIP\_and\_NN\_Search\_Problems/BIBREF62\_5b0a88bdec473552c6a386cd94fdac53c74b79a8.pdf Title: On Symmetric and Asymmetric LSHs for Inner Product Search Chunk of text: For an integer parameter m, and real valued parameters 0 < U < 1 and r > 0, consider the following pair of mappings: P(x) = [Ux; kUxk 2 ; kUxk 4 ; . . . ; kUxk 2m ] Q(y) = [y; 1/2; 1/2; . . .; 1/2], (6) combined with the standard L2 hash function h L2 a,b(x) = a ⊤x + b r (7) where a ∼ N (0, I) is a spherical multi-Gaussian random vector, b ∼ U(0, r) is a uniformly distributed random variable on [0, r]. The alphabet Γ used is the integers, the intermediate space is Z = R d+m and the asymmetric hash L2-ALSH(SL), parameterized by m, U and r, is then given by (f(x), g(q)) = (h L2 a,b(P(x)), hL2 a,b(Q(q))).

[REF1] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-MIP\_and\_NN\_Search\_Problems/BIBREF62\_5b0a88bdec473552c6a386cd94fdac53c74b79a8.pdf Title: On Symmetric and Asymmetric LSHs for Inner Product Search Chunk of text: For L2-ALSH(SL) and SIGN-ALSH(SL), for each desired threshold S and ratio c, one can optimize over the parameters m and U, and for L2-ALSH(SL) also r, to find the hash with the best ρ. This is a non-convex optimization problem and Shrivastava and Li (2014a) suggest using grid search to find a bound on the optimal ρ. We followed the procedure, and grid, as suggested by Shrivastava and Li (2014a) 3 . For SIMPLE-LSH no parameters need to be tuned, and for each S, c the hashing quality is given by Theorem 5.3. In Figure 1 we compare the optimal hashing quality ρ for the three methods, for different values of S and c. It is clear that the SIMPLE-LSH dominates the other methods. 4.4.

[REF2] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-MIP\_and\_NN\_Search\_Problems/BIBREF62\_5b0a88bdec473552c6a386cd94fdac53c74b79a8.pdf Title: On Symmetric and Asymmetric LSHs for Inner Product Search Chunk of text: Here, we study the transformation as part of an LSH scheme, investigate its theoretical properties, and compare it to LS-ALSH(SL). 2. Locality Sensitive Hashing A hash of a set Z of objects is a random mapping from Z to some alphabet Γ, i.e. a distribution over functions h : Z → Γ. The hash is sometimes thought of as a “family” of functions, where the distribution over the family is implicit. When studying hashes, we usually study the behavior when comparing any two points x, y ∈ Z. However, for our study here, it will be important for us to make different assumptions about x and y—e.g., we will want to assume w.l.o.g. that queries are normalized but will not be able to make the same assumptions on database vectors. To this end, we define what it means for a hash to be an LSH over a pair of constrained subspaces X, Y ⊆ Z. Given a similarity function sim : Z × Z → R, such as inner product similarity sim(x, y) = x ⊤y, an LSH is defined as follows1 : Definition 1 (Locality Sensitive Hashing (LSH)).

[REF3] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-MIP\_and\_NN\_Search\_Problems/BIBREF62\_5b0a88bdec473552c6a386cd94fdac53c74b79a8.pdf Title: On Symmetric and Asymmetric LSHs for Inner Product Search Chunk of text: • When queries and database vectors are bounded but not normalized, a symmetric LSH is not possible, but a universal asymmetric LSH is. Here we see the power of asymmetry. This corrects the view of Shrivastava and Li (2014a), who used the nonexistence of a symmetric LSH over R d to motivate an asymmetric LSH when queries are normalized and database vectors are bounded, even though we now see that in these two settings there is actually no advantage to asymmetry. In the third setting, where an asymmetric hash is indeed needed, the hashes suggested by Shrivastava and Li (2014a;b) are not ALSH, and a different asymmetric hash is required (which we provide). Furthermore, even in the MIPS setting when queries are normalized (the second setting), the asymmetric hashes suggested by Shrivastava and Li (2014a;b) are not universal and require tuning parameters specific to S, c, in contrast to SIMPLE-LSH which is symmetric, parameter-free and universal. It is important to emphasize that even though in the MIPS setting an asymmetric hash, as we define here, is not needed, an asymmetric view of the problem is required.

[REF4] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-MIP\_and\_NN\_Search\_Problems/BIBREF62\_5b0a88bdec473552c6a386cd94fdac53c74b79a8.pdf Title: On Symmetric and Asymmetric LSHs for Inner Product Search Chunk of text: SIMPLE-LSH does not require any parameters. As can be seen in the Figures, SIMPLE-LSH shows a dramatic empirical improvement over L2-ALSH(SL). Following the presentation of SIMPLE-LSH and the comparison with L2-ALSH(SL), Shrivastava and Li (2014b) suggested the modified hash SIGN-ALSH(SL), which is based on random projections, as is SIMPLE-LSH, but with an asymmetric transform similar to that in L2-ALSH(SL). PerhapsOn Symmetric and Asymmetric LSHs for Inner Product Search 0 0.2 0.4 0.6 0.8 1 0 0.1 0.2 0.3 0.4 Precision Top 10, K = 64 SIMPLE−LSH L2−ALSH(SL) SIGN−ALSH(SL),m=2 SIGN−ALSH(SL),m=3 0 0.2 0.4 0.6 0.8 1 0 0.1 0.2 0.3 0.4 0.5 Top 10, K = 128 SIMPLE−LSH L2−ALSH(SL) SIGN−ALSH(SL),m=2 SIGN−ALSH(SL),m=3 0 0.2 0.4 0.6 0.8 1 0 0.2 0.4 0.6 0.8 Top 10, K = 256 SIMPLE−LSH L2−ALSH(SL) SIGN−ALSH(SL),m=2 SIGN−ALSH(SL),m=3 0 0.2 0.4 0.6 0.8 1 0 0.2 0.4 0.6 0.8 Top 10, K = 512 SIMPLE−LSH L2−ALSH(SL)

[REF5] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-MIP\_and\_NN\_Search\_Problems/BIBREF62\_5b0a88bdec473552c6a386cd94fdac53c74b79a8.pdf Title: On Symmetric and Asymmetric LSHs for Inner Product Search Chunk of text: Therefore, we have q ⊤x = cS and q ⊤y = S. However, since q = x, Ph(h(q) = h(x)) = 1 ≤ p2 < p1 = Ph(h(q) = h(y)) ≤ 1 and we get a contradiction. 5.2.

[REF6] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-MIP\_and\_NN\_Search\_Problems/BIBREF62\_5b0a88bdec473552c6a386cd94fdac53c74b79a8.pdf Title: On Symmetric and Asymmetric LSHs for Inner Product Search Chunk of text: = 1 − cos−1 (q ⊤x) π . As in the proof of Theorem 4.2, monotonicity of 1 − cos−1 (x) π establishes the desired ALSH properties. Shrivastava and Li (2015) also showed how a modification of SIMPLE-ALSH can be used for searching similarity measures such as set containment and weighted Jaccard similarity. 6. Conclusion We provide a complete characterization of when symmetric and asymmetric LSH are possible for inner product similarity: • Over R d , no symmetric nor asymmetric LSH is possible. • For the MIPS setting, with normalized queries kqk = 1 and bounded database vectors kxk ≤ 1, a universal symmetric LSH is possible. • When queries and database vectors are bounded but not normalized, a symmetric LSH is not possible, but a universal asymmetric LSH is.

[REF7] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-MIP\_and\_NN\_Search\_Problems/BIBREF62\_5b0a88bdec473552c6a386cd94fdac53c74b79a8.pdf Title: On Symmetric and Asymmetric LSHs for Inner Product Search Chunk of text: Furthermore, even in the MIPS setting when queries are normalized (the second setting), the asymmetric hashes suggested by Shrivastava and Li (2014a;b) are not universal and require tuning parameters specific to S, c, in contrast to SIMPLE-LSH which is symmetric, parameter-free and universal. It is important to emphasize that even though in the MIPS setting an asymmetric hash, as we define here, is not needed, an asymmetric view of the problem is required. In particular, to use a symmetric hash, one must normalize the queries but not the database vectors, which can legitimately be viewed as an asymmetric operation which is part of the����On Symmetric and Asymmetric LSHs for Inner Product Search hash (though then the hash would not be, strictly speaking, an ALSH). In this regard Shrivastava and Li (2014a) do indeed successfully identify the need for an asymmetric view of MIPS, and provide the first practical ALSH for the problem. ACKNOWLEDGMENTS This research was partially funded by NSF award IIS1302662.

[REF8] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-MIP\_and\_NN\_Search\_Problems/BIBREF62\_5b0a88bdec473552c6a386cd94fdac53c74b79a8.pdf Title: On Symmetric and Asymmetric LSHs for Inner Product Search Chunk of text: Introduction Following Shrivastava and Li (2014a), we consider the problem of Maximum Inner Product Search (MIPS): given a collection of “database” vectors S ⊂ R d and a query q ∈ R d , find a data vector maximizing the inner product with the query: p = arg max x∈S q ⊤x (1) MIPS problems of the form (1) arise, e.g. when using matrix-factorization based recommendation systems (Koren et al., 2009; Srebro et al., 2005; Cremonesi et al., 2010), in multi-class prediction (Dean et al., 2013; Jain et al., 2009) and structural SVM (Joachims, 2006; Joachims et al., 2009) problems and in vision problems when scoring filters based on their activations (Dean et al., 2013) (see Shrivastava and Li, 2014a, for more about MIPS). In order to efficiently find approximate MIPS solutions, Shrivastava and Li (2014a) suggest constructing a Locality Sensitive Hash (LSH) for inner product “similarity”. Proceedings of the 31 st International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s). Locality Sensitive Hashing (Indyk and Motwani, 1998) is a popular tool for approximate nearest neighbor search and is also widely used in other settings (Gionis et al., 1999; Datar et al., 2004; Charikar, 2002).

[REF9] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-MIP\_and\_NN\_Search\_Problems/BIBREF62\_5b0a88bdec473552c6a386cd94fdac53c74b79a8.pdf Title: On Symmetric and Asymmetric LSHs for Inner Product Search Chunk of text: ≈ P[h(x) = g(y)] (Neyshabur et al., 2013; 2014). Neyshabur et al. showed that even when the similarity sim(x, y) is entirely symmetric, asymmetry in the hash may enable obtaining an LSH when a symmetric LSH is not possible, or enable obtaining a much better LSH yielding shorter and more accurate hashes. Several tree-based methods have also been proposed for inner product search (Ram and Gray, 2012; Koenigstein et al., 2012; Curtin et al., 2013). Shrivastava and Li (2014a) argue that tree-based methods, such as cone trees, are impractical in high dimensions while the performance of LSH-based methods is in a way independent of dimension of the data. Although the exact regimes under which LSH-based methods are superior to tree-based methods and vice versa are not fully established yet, the goal of this paper is to analyze different LSH methods and compare them with each other, rather than comparing to tree-based methods, so as to understand which LSH to use and why, in those regimes where tree-based methods are not practical. Considering MIPS, Shrivastava and Li (2014a) argue that there is no symmetric LSH for inner product similarity, and propose two distinct mappings, one of database objects and the other for queries, which yields an asymmetric LSH for MIPS.

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

Title: Retrieval Architectures and Vector Search - Locality sensitive hashing approaches

Locality sensitive hashing (LSH) is a technique used in information retrieval to solve the nearest neighbor problem. LSH function families have the property that objects that are close to each other have a higher probability of colliding than objects that are far apart [REF2]. LSH solves the decision version of the nearest neighbor problem, while kd-tree solves the optimization version. However, the reduction overhead in kd-tree increases the running time [REF0].

To achieve high search accuracy, the LSH method needs to use multiple hash tables to produce a good candidate set. The basic LSH method requires over a hundred and sometimes several hundred hash tables to achieve good search accuracy for high-dimensional datasets [REF3]. This approach does not satisfy the space-efficiency requirement. To address this issue, an entropy-based LSH method has been proposed, which generates randomly perturbed objects near the query object and queries them in addition to the query object [REF3]. This method reduces the space requirement of the basic LSH method while still achieving good search accuracy [REF3].

Another approach to improve the efficiency of LSH is the multi-probe LSH method. This method effectively reduces the space requirement while achieving the desired search quality with more probes [REF1]. Experimental studies have shown that multi-probe LSH achieves similar search quality for different values of K, the number of nearest neighbors [REF1]. By using more probes, the multi-probe LSH method can effectively reduce the space requirement without compromising search quality [REF1].

The memory requirement for LSH algorithms depends on the number of data objects and the number of hash tables used. In typical scenarios, the memory requirement can be reduced by storing the hash values implicitly instead of explicitly concatenating them [REF6]. This reduces the memory requirement per data point, making the LSH algorithm more space-efficient [REF6].

LSH has been widely studied in the context of similarity search in high-dimensional spaces. An ideal indexing scheme for similarity search should be accurate, time-efficient, and space-efficient [REF7]. LSH techniques aim to achieve these properties by using hash functions that increase the probability of collision for similar objects and reduce the space requirement for indexing [REF2] [REF3] [REF6] [REF7].

In summary, LSH approaches provide efficient solutions to the nearest neighbor problem in information retrieval. These approaches use hash functions to increase the probability of collision for similar objects and reduce the space requirement for indexing. The multi-probe LSH method and the entropy-based LSH method are two variations that improve the efficiency and space requirement of LSH algorithms. By using multiple hash tables and more probes, these methods achieve high search accuracy while reducing the space requirement.

References given to GPT:

that is much larger than the intended approximation. Although the resulting algorithm would provide very weak guarantee on the quality of the returned neighbor, typically the actual error is much smaller than the guarantee. 7.

[REF1] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Locality\_sensitive\_hashing\_approaches/BIBREF66\_9ed960374381062d85d3944182a539c1d00f7703.pdf Title: Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search Chunk of text: However, for a given K, the multi-probe LSH method can effectively reduce the space requirement while achieving desired search quality with more probes. 7. RELATED WORK 0.7 0.75 0.8 0.85 0.9 0.95 1 0 50 100 150 200 250 300 350 400 450 500 Recall Number of Probes image K=20 K=60 K=100 0.7 0.75 0.8 0.85 0.9 0.95 1 0 50 100 150 200 250 300 350 400 450 500 Recall Number of Probes audio K=20 K=60 K=100 Figure 10: Recall of multi-probe LSH for different K (number of nearest neighbors): multi-probe LSH achieves similar search quality for different K values. method image audio recall C/N (%) recall C/N (%) basic 0.96 4.4 0.94 6.3 entropy 0.96 4.9 0.94 6.8 multi-probe 0.96 5.1 0.94 7.1 basic 0.93 3.3 0.92 5.7 entropy 0.93 3.9 0.92 5.9 multi-probe 0.93 4.1 0.92 6.0 basic 0.90 2.6 0.90 5.0 entropy 0.90 3.1 0.90 5.6 multi-probe 0.90 3.0 0.90 5.3 Table 4: Percentage of objects examined using different LSH methods (C is candidate set size, N is dataset size): multi-probe LSH has similar filter ratio as other LSH methods. The similarity search problem is closely related to the nearest neighbor search problem, which has been studied extensively.

[REF2] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Locality\_sensitive\_hashing\_approaches/BIBREF66\_9ed960374381062d85d3944182a539c1d00f7703.pdf Title: Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search Chunk of text: The notion of locality sensitive hashing (LSH) was first introduced by Indyk and Motwani in . LSH function families have the property that objects that are close to each other have a higher probability of colliding than objects that are far apart. Specifically, let S be the domain of objects, and D be the distance measure between objects. Definition 1. A function family H = {h : S → U} is called (r, cr, p1, p2)-sensitive for D if for any q, p ∈ S • If D(q, p) ≤ r then P rH[h(q) = h(p)] ≥ p1, • If D(q, p) > cr then P rH[h(q) = h(p)] ≤ p2.

[REF3] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Locality\_sensitive\_hashing\_approaches/BIBREF66\_9ed960374381062d85d3944182a539c1d00f7703.pdf Title: Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search Chunk of text: To achieve high search accuracy, the LSH method needs to use multiple hash tables to produce a good candidate set. Experimental studies show that this basic LSH method needs over a hundred and sometimes several hundred hash tables to achieve good search accuracy for high-dimensional datasets. Since the size of each hash table is proportional to the number of data objects, the basic approach does not satisfy the spaceefficiency requirement. In a recent theoretical study , Panigrahy proposed an entropy-based LSH method that generates randomly “perturbed” objects near the query object, queries them in addi-tion to the query object, and returns the union of all results as the candidate set. The intention of the method is to trade time for space requirements. To explore the practicality of this approach, we have implemented it and conducted an experimental study. We found that although the entropybased method can reduce the space requirement of the basic LSH method, significant improvements are possible.

[REF4] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Locality\_sensitive\_hashing\_approaches/BIBREF65\_3f1e54ed3bd801766e1897d53a9fc962524dd3c2.pdf Title: Locality-Sensitive Hashing Scheme Based on p-Stable Distributions Chunk of text: -NN under measure D which uses O(dn + n1+) space, with query time dominated by O(n) distance computations, and O(n log1=p2 n) evaluations of hash functions from H, where = ln 1=p1 ln 1=p2 . 3. OUR LSH SCHEME In this section, we present a LSH family based on p-stable distributions, that works for all p 2 (0; 2℄. Since we consider points in l d p, without loss of generality we can consider R = 1, which we assume from now on. 3.1 p-stable distributions Stable distributions are defined as limits of normalized sums of independent identically distributed variables (an alternate definition follows).

[REF5] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Locality\_sensitive\_hashing\_approaches/BIBREF66\_9ed960374381062d85d3944182a539c1d00f7703.pdf Title: Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search Chunk of text: This scheme works as follows. Assuming we know the distance Rp from the nearest neighbor p to the query q. In principle, for every hash bucket, we can compute the probability that p lies in that hash bucket (call this the success probability of the hash bucket). Note that this distribution depends only on the distance Rp. Given this information, it would make sense to query the hash buckets which have the highest success probabilities. However, performing this calculation is cumbersome. Instead, Panigrahy proposes a clever way to sample buckets from the distribution given by these probabilities.

[REF6] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Locality\_sensitive\_hashing\_approaches/BIBREF65\_3f1e54ed3bd801766e1897d53a9fc962524dd3c2.pdf Title: Locality-Sensitive Hashing Scheme Based on p-Stable Distributions Chunk of text: Memory Requirement: The memory requirement for our algorithm equals the memory to store the data points themselves and the memory required to store the hash tables. From our experiments, typical values of k and l are 10 and 30 respectively. If we insert each point in the hash tables along with their hash values and a pointer to the data point itself, it will require l (k + 1) words (int) of memory, which for our typical k; l values evaluates to 330 words. We can reduce the memory requirement by not storing the hash value explicitly as concatenation of k projections, but instead hash these k values in turn to get a single word for the hash. This would reduce the memory requirement to l 2, i.e. 60 words per data point. If the data points belong to a high dimensional space (e.g., with 500 dimension or more), then the overhead of maintaining the hash table is not much (around 12% with the optimization above) as compared to storing the points themselves.

[REF7] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Locality\_sensitive\_hashing\_approaches/BIBREF66\_9ed960374381062d85d3944182a539c1d00f7703.pdf Title: Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search Chunk of text: Copyright 2007 VLDB Endowment, ACM 978-1-59593-649-3/07/09. An ideal indexing scheme for similarity search should have the following properties: • Accurate: A query operation should return desired results that are very close to those of the brute-force, linear-scan approach. • Time efficient: A query operation should take O(1) or O(log N) time where N is the number of data objects in the dataset. • Space efficient: An index should require a very small amount of space, ideally linear in the dataset size, not much larger than the raw data representation.

[REF8] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Locality\_sensitive\_hashing\_approaches/BIBREF66\_9ed960374381062d85d3944182a539c1d00f7703.pdf Title: Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search Chunk of text: INTRODUCTION Similarity search in high-dimensional spaces has become increasingly important in databases, data mining, and search engines, particularly for content-based search of feature-rich data such as audio recordings, digital photos, digital videos, and other sensor data. Since feature-rich data objects are typically represented as high-dimensional feature vectors, similarity search is usually implemented as K-Nearest Neighbor (KNN) or Approximate Nearest Neighbors (ANN) search in high-dimensional feature-vector space. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires a fee and/or special permission from the publisher, ACM. VLDB ‘07, September 23-28, 2007, Vienna, Austria. Copyright 2007 VLDB Endowment, ACM 978-1-59593-649-3/07/09. An ideal indexing scheme for similarity search should have the following properties: •

[REF9] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Locality\_sensitive\_hashing\_approaches/BIBREF66\_9ed960374381062d85d3944182a539c1d00f7703.pdf Title: Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search Chunk of text: However, performing this calculation is cumbersome. Instead, Panigrahy proposes a clever way to sample buckets from the distribution given by these probabilities. Each time, a random point p 0 at distance Rp from q is generated and the bucket that p 0 is hashed to is checked. This ensures that buckets are sampled with exactly the right probabilities. Performing this sampling multiple times will ensure that all the buckets with high success probabilities are probed. However, this approach has some drawbacks: the sampling process is inefficient because perturbing points and computing their hash values are slow, and it will inevitably generate duplicate buckets. In particular, buckets with high success probability will be generated multiple times and much of the computation is wasteful.

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

Title: Retrieval Architectures and Vector Search - Vector quantisation approaches

Vector quantisation is a widely used technique in information retrieval systems for efficient representation and retrieval of high-dimensional data. It involves mapping data points to a set of centroids in a codebook, thereby reducing the dimensionality of the data. In this section, we discuss retrieval architectures and vector search techniques based on vector quantisation approaches.

One approach to vector quantisation is the use of product quantisers, where the quantisation function associated with each component may be different [REF3]. Product quantisers have the advantage of producing a large set of centroids from several small sets of centroids, which are associated with subquantisers. This approach allows for efficient representation of data distributions and reduces the complexity of learning the quantiser [REF3]. However, the explicit storage of the codebook may not be efficient [REF3].

Another vector quantisation approach is the use of inverted file with asymmetric distance computation (IVFADC) indexing system [REF4]. In IVFADC, a query vector is assigned to multiple indexes, corresponding to the nearest neighbors of the query vector in the codebook. This multiple assignment strategy allows for efficient search by scanning the inverted lists associated with the assigned indexes [REF4]. IVFADC has been shown to provide good search efficiency, especially for larger database sizes [REF7].

The design of vector quantisers can be guided by techniques such as variable-rate coding and entropy-constrained quantisation [REF1]. Variable-rate vector quantisers allow for the allocation of different numbers of bits to different components of the vector, providing a simple and exact solution to the bit allocation problem [REF1]. Entropy-constrained vector quantisers, on the other hand, trade off compression performance with added complexity, but can achieve excellent compression ratios [REF1].

In recent years, there has been a focus on developing vector quantisation methods that are suitable for large-scale applications [REF9]. These methods aim to limit the memory usage while still providing efficient retrieval of large amounts of data. For example, Torralba et al. proposed a method that represents images using a single global descriptor and maps it to a short binary code [REF9]. The search for nearest neighbors is then approximated using Hamming distances between the codes [REF9]. Spectral hashing (SH) has also been shown to outperform other binary codes in large-scale retrieval tasks [REF9].

Overall, vector quantisation approaches offer efficient retrieval architectures and vector search techniques for high-dimensional data. Product quantisers and IVFADC indexing systems provide effective ways to represent and search for data points in large databases. Additionally, techniques such as variable-rate coding and entropy-constrained quantisation contribute to the design of efficient vector quantisers. The development of methods suitable for large-scale applications further enhances the applicability of vector quantisation in neural information retrieval systems.

[REF1] - [REF9]

References given to GPT:

[REF0] - 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 query times per descriptor are shown on Figure 11. The cost of the extra quantization step required by IVFADC appears clearly for small database sizes. For larger scales, the distance computation with the database vectors become preponderant. The processing that is applied to each element of the inverted lists is13 method parameters search average number of recall@100 time (ms) code comparisons SDC 16.8 1 000 991 0.446 ADC 17.2 1 000 991 0.652 IVFADC k ′= 1 024, w=1 1.5 1 947 0.308 k ′= 1 024, w=8 8.8 27 818 0.682 k ′= 1 024, w=64 65.9 101 158 0.744 k ′= 8 192, w=1 3.8 361 0.240 k ′= 8 192, w=8 10.2 2 709 0.516 k ′= 8 192, w=64 65.3 19 101 0.610 SH 22.7 1 000 991 0.132 TABLE V GIST DATASET (500 QUERIES): SEARCH TIMINGS FOR 64-BIT CODES AND DIFFERENT METHODS. WE HAVE USED m=8 AND k ∗ =256 FOR SDC, ADC AND IVFADC. 0 0.5 1 1.5 2 2.5 3 3.5 10M 100M 1G search time (ms/vector) database size HE IVFADC Fig. 11.

[REF1] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Vector\_quantisation\_approaches/BIBREF67\_c564aa7639a08c280423489e52b6e32055c9aa7f.pdf Title: Vector Quantization and Signal Compression Chunk of text: Much of Chapter 17 consists of taking advantage of the similarities of variable-rate vector quantizers and decision trees for statistical pattern classification in order to develop coder design algorithms for unbalanced tree-structured vector quantizers. Methods of growing and pruning such tree-structured coders are detailed. As vector quantizers can be used in conjunction with entropy coding to obtain even further compression at the expense of the added complication and the necessity of variable-rate coding, the design of vector quantizers specifically for such application is considered. Such entropy-constrained vector quantizers are seen to provide excellent compression if one is willing to pay the price. The techniques for designing variable-rate vector quantizers are shown to provide a simple and exact solution to the bit allocation problem introduced in Chapter 8 and important for a variety of vector quantizer structures, including classified and transform vector quantizers. Instructional Use This book is intended both as a reference text and for use in a graduate Electrical Engineering course on quantization and signal compression. Its self-contained development of prerequisites, traditional techniques, and vector quantization together with its extensive citations of the literature make xx PREFACE the book useful for a general and thorough introduction to the field or for occasional searches for descriptions of a particular technique or the relative merits of different approaches.

[REF2] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Vector\_quantisation\_approaches/BIBREF67\_c564aa7639a08c280423489e52b6e32055c9aa7f.pdf Title: Vector Quantization and Signal Compression Chunk of text: 12.5 Classified VQ ..... .. 12.6 Transform VQ . . .... . 12.7 Product Code Techniques 12.8 Partitioned VQ .. 12.9 Mean-Removed VQ 12.10 Shape-Gain VQ .. 12.11 Multistage VQ ... 12.12 Constrained Storage VQ 12.13 Hierarchical and Multiresolution VQ 12.14 Nonlinear Interpolative VQ .... 12.15 Lattice Codebook VQ . . . . . . . 12.16 Fast Nearest Neighbor Encoding. 12.17 Problems . . . . . . . . . . . . 13 Predictive Vector Quantization 13.1 Introduction ......... . 13.2 Predictive Vector Quantization 13.3 Vector Linear Prediction ....

[REF3] - 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: Then the product quantizer turns out to be a scalar quantizer, where the quantization function associated with each component may be different. The strength of a product quantizer is to produce a large set of centroids from several small sets of centroids: those associated with the subquantizers. When learning the subquantizers using Lloyd’s algorithm, a limited number of vectors is used, but the codebook is, to some extent, still adapted to the data distribution to represent. The complexity of learning the quantizer is m times the complexity of performing k-means clustering with k ∗ centroids of dimension D∗�����4 memory usage assignment complexity k-means k D k D HKM bf bf−1 (k − 1) D l D product k-means m k∗ D ∗ = k 1/m D m k∗ D ∗ = k 1/m D TABLE I MEMORY USAGE OF THE CODEBOOK AND ASSIGNMENT COMPLEXITY FOR DIFFERENT QUANTIZERS. HKM IS PARAMETRIZED BY TREE HEIGHT l AND THE BRANCHING FACTOR bf . Storing the codebook C explicitly is not efficient.

[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: Overview of the inverted file with asymmetric distance computation (IVFADC) indexing system. Top: insertion of a vector. Bottom: search. address this problem, we use the multiple assignment strategy of . The query x is assigned to w indexes instead of only one, which correspond to the w nearest neighbors of x in the codebook of qc. All the corresponding inverted lists are scanned. Multiple assignment is not applied to database vectors, as this would increase the memory usage.

[REF5] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Vector\_quantisation\_approaches/BIBREF67\_c564aa7639a08c280423489e52b6e32055c9aa7f.pdf Title: Vector Quantization and Signal Compression Chunk of text: The development of useful design algorithms and coding structures began in the late 1970s and interest in vector quantization expanded rapidly in the 1980s. Prior to that time digital signal processing circuitry was not fast enough and the memories were not large enough to use vector coding techniques in real time and there was little interest in design algorithms for such codes. The rapid advance in digital signal processor chips in the past decade made possible low cost implementations of such algorithms that would have been totally infeasible in the 1970s. During the past ten years, vector quantization has proved a valuable coding technique in a variety of applications, especially in voice and image coding. This is because of its simple structure, its ability to trade ever cheaper memory for often expensive computation, and the often serendipitous structural properties of the codes designed by iterative clustering algorithms. As an example of the desirable struct ural properties of vector quantizers, suitably designed tree-structured codes are nested and are naturally optimized for progressive transmission applications where one progressively improves a signal (such as an image) as more bits arrive. Another example is the ability of clustering algorithms used to design vector quantizers to enhance certain features of the original signal such as small tumors in a medici:il image.

[REF6] - 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: SIFT: quantization error associated with the parameters m and k ∗ . symmetric case asymmetric case Fig. 2. Illustration of the symmetric and asymmetric distance computation. The distance d(x, y) is estimated with either the distance d(q(x), q(y)) (left) or the distance d(x, q(y)) (right). The mean squared error on the distance is on average bounded by the quantization error.

[REF7] - 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: In this work, the authors propose a hierarchical quantizer to efficiently assign descriptors to one million centroids. F. Large-scale experiments To evaluate the search efficiency of the product quantizer method on larger datasets we extracted about 2 billion SIFT descriptors from one million images. Search is performed with 30 000 query descriptors from ten images. We compared the IVFADC and HE methods with similar parameters. In particular, the amount of memory that is scanned for each method and the cost of the coarse quantization are the same. The query times per descriptor are shown on Figure 11. The cost of the extra quantization step required by IVFADC appears clearly for small database sizes.

[REF8] - 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: A. Vector quantization Quantization is a destructive process which has been extensively studied in information theory . Its purpose is to reduce the cardinality of the representation space, in particular when the input data is real-valued. Formally, a quantizer is a function q mapping a Ddimensional vector x ∈ R D to a vector q(x) ∈ C = {ci ;i ∈ I}, where the index set I is from now on assumed to be finite: I = 0 . . . k − 1. The reproduction values ci are called centroids. The set of reproduction values C is the codebook of size k. The set Vi of vectors mapped to a given index i is referred to as a (Voronoi) cell, and defined as Vi , {x ∈ R D : q(x) = ci}.

[REF9] - 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: Only recently, researchers came up with methods limiting the memory usage. This is a key criterion for problems involving large amounts of data , i.e., in large-scale scene recognition , where millions to billions of images have to be indexed. In , Torralba et al. represent an image by a single global GIST descriptor which is mapped to a short binary code. When no supervision is used, this mapping is learned such that the neighborhood in the embedded space defined by the Hamming distance reflects the neighborhood in the Euclidean space of the original features. The search of the Euclidean nearest neighbors is then approximated by the search of the nearest neighbors in terms of Hamming distances between codes. In , spectral hashing (SH) is shown to outperform the binary codes generated by the restricted Boltzmann machine , boosting and LSH.

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

Title: Retrieval Architectures and Vector Search - Graph approaches

Graph-based approaches have gained significant attention in the field of neural information retrieval due to their ability to capture complex relationships and dependencies among data points. These approaches leverage graph structures to model the similarity and connectivity between documents, enabling efficient and effective retrieval of relevant information. In this section, we discuss retrieval architectures and vector search techniques based on graph approaches.

One popular graph-based retrieval architecture is the Navigable Small World (NSW) graph [REF0]. NSW is a multilayer graph that organizes data points in a hierarchical manner, allowing for efficient search and retrieval operations. The construction of the NSW graph involves establishing connections between elements based on their similarity, with the goal of creating a small-world network topology. The NSW graph exhibits logarithmic scalability and provides an effective solution for approximate k-nearest neighbor (K-NNG) search problems [REF2].

The construction of the NSW graph involves several steps. Initially, an enter point is selected as the starting point for the graph construction. The graph is then built layer by layer, with each layer representing a different level of similarity. The connections between elements are established based on their proximity in the graph. The NSW graph construction algorithm employs a greedy search algorithm to approximate the K-NNG problem [REF2]. This algorithm iteratively searches for the nearest elements to a given query, expanding the search to lower layers of the graph. The connections between elements are updated based on the search results, ensuring the graph's connectivity and accuracy [REF0].

Another graph-based approach for vector search is the decentralized algorithm [REF9]. This algorithm utilizes lattice distance to determine the proximity between data points. In each step, the current message-holder selects a contact that is as close to the target as possible. The algorithm operates in phases, with each phase representing a different level of lattice distance. The decentralized algorithm efficiently explores the graph structure, enabling effective vector search operations [REF9].

Graph-based retrieval architectures offer several advantages in neural information retrieval. Firstly, they capture complex relationships and dependencies among data points, allowing for more accurate and context-aware retrieval. Secondly, these architectures provide efficient search and retrieval operations, enabling real-time processing of large-scale datasets. Lastly, graph-based approaches can be easily parallelized and distributed across multiple machines, facilitating scalability and performance improvements [REF3].

In summary, retrieval architectures and vector search techniques based on graph approaches have shown promising results in neural information retrieval. The NSW graph and the decentralized algorithm are examples of effective graph-based approaches that leverage the connectivity and similarity relationships between data points. These approaches provide efficient and accurate retrieval operations, making them suitable for various applications in the field of information retrieval.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Graph\_approaches/BIBREF72\_699a2e3b653c69aff5cf7a9923793b974f8ca164.pdf Title: Efficient and Robust Approximate Nearest Neighbor Search using Hierarchical Navigable Small World Graphs Chunk of text: The distinctions from NSW (along with some queue optimizations) are: 1) the enter point is a fixed parameter; 2) instead of changing the number of multi-searches, the quality of the search is controlled by a different parameter ef (which was set to K in NSW ). Algorithm 1 INSERT(hnsw, q, M, Mmax, efConstruction, mL) Input: multilayer graph hnsw, new element q, number of established connections M, maximum number of connections for each element per layer Mmax, size of the dynamic candidate list efConstruction, normalization factor for level generation mL Output: update hnsw inserting element q 1 W ← ∅ // list for the currently found nearest elements 2 ep ← get enter point for hnsw 3 L ← level of ep // top layer for hnsw 4 l ← ⌊-ln(unif(0..1))∙mL⌋ // new element’s level 5 for lc ← L … l+1 6 W ← SEARCH-LAYER(q, ep, ef=1, lc) 7 ep ← get the nearest element from W to q 8 for lc ← min(L, l) … 0 9 W ← SEARCH-LAYER(q, ep, efConstruction, lc) 10 neighbors ← SELECT-NEIGHBORS(q, W, M, lc) // alg. 3 or alg. 4 11 add bidirectionall connectionts from neighbors to q at layer lc 12 for each e ∈ neighbors // shrink connections if needed 13 eConn ← neighbourhood(e) at layer lc 14 if │eConn│ > Mmax // shrink connections of e // if lc = 0 then Mmax = Mmax0 15 eNewConn ← SELECT-NEIGHBORS(e, eConn, Mmax, lc) // alg. 3 or alg. 4 16 set neighbourhood(e) at layer lc to eNewConn 17 ep ← W 18 if l > L 19 set enter point for hnsw to q Algorithm 2 SEARCH-LAYER(q, ep, ef, lc)

[REF1] - 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 PROPOSED METHOD 2.1 Notations and Background Let V be a dataset of size N = |V |, and let σ : V ×V → R be a similarity measure. For each v ∈ V , let BK(v) be v’s K-NN, i.e. the K objects in V (other than v) most similar to v. Let RK(v) = {u ∈ V | v ∈ BK(u)} be v’s reverse K-NN. In the algorithm, we use B[v] and R[v] to store the approximation of BK(v) and RK(v), together with the similarity values, and let B[v] = B[v]∪ R[v], referred to as the general neighbors of v. B[v] is organized as a heap, so updates cost O(log K). We are particularly interested in the case when V is a metric space with a distance metric d : V × V → [0, +∞) for which more specific analysis can be done. Since smaller distance means higher similarity, we simply let σ = −d. For any r ∈ [0, +∞), the r-ball around v ∈ V is defined as Br(v) = {u ∈ V | d(u, v) ≤ r}.

[REF2] - 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 Graphs Chunk of text: Graphs with logarithmic scalability of the greedy search algorithm are called navigable small world graphs, they are well known in Euclidean spaces . Note that the small world models (not navigable small world) like do not have this feature. Even though there are short paths in the graph, the greedy algorithm do no tend to find them, in the end having a power law search complexity. Solutions for constructing a navigational small world graphs were proposed for general spaces but they are usually more complex, requiring sampling, iterations, rewiring etc. [11–14]. We show that the small world navigation property can be achieved with a much simpler technique even without prior knowledge of internal structure of a metric space (e.g. dimensionality or data density distribution). In this paper we present a simple algorithm for the data structure construction based on a navigable small world network topology with a graph GðV; EÞ, which uses the greedy search algorithm for the approximate k-nearest neighbor search problem.

[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: Paredes et al. proposed two methods for K-NNG construction in general metric spaces with low empirical complexity, but both require a global data structure and are hard to parallelize across machines. Efficient methods for l2 distance have been developed based on recursive data partitioning and space filling curves , but they do not naturally generalize to other distance metrics or general similarity measures. Indexing data for K-NN search is a closely related open problem that has been extensively studied. A K-NNG can be constructed simply by repetitively invoking K-NN search for each object in the dataset. Various tree-based data structures are designed for both general metric space and Euclidean space

[REF4] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Graph\_approaches/BIBREF72\_699a2e3b653c69aff5cf7a9923793b974f8ca164.pdf Title: Efficient and Robust Approximate Nearest Neighbor Search using Hierarchical Navigable Small World Graphs Chunk of text: Product quantization K-ANNS algorithms [10-17] are considered as the state-of-the-art on billion scale datasets since they can efficiently compress stored data, allowing modest RAM usage while achieving millisecond search times on modern CPUs. To compare the performance of Hierarchical NSW against PQ algorithms we used the facebook Faiss library8 as the baseline (a new library with state-of-the-art PQ algorithms [12, 15] implementations, released after the current manuscript was submitted) compiled with the OpenBLAS backend. The tests where done for a 200M subset of 1B SIFT dataset on a 4X Xeon E5-4650 v2 server with 128Gb of RAM. The ann-benchmark testbed was not feasible for these experiments because of its reliance on 32-bit floating point format (requiring more than 100 Gb just to store the data). To get the results for Faiss PQ algorithms we have utilized built-in scripts with the parameters from Faiss wiki9 .

[REF5] - 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: Assume we already have an approximate K-NNG B, and for every v ∈ V , let B ′ [v] = S v′∈B[v] B[v ′ ] be the set of points we explore trying to improve B. If the accuracy of B is reasonably good, such that for certain fixed radius r, for all v ∈ V , B[v] contains K neighbors that are uniformly distributed in Br(v), then assuming independence of certain events and that k ≪ |Br/2(v)|, we can conclude that B ′ [v] is likely to contain K neighbors in Br/2(v). In other words, we expect to halve the maximal distance to the set of approximate K nearest neighbors by exploring B ′ [v] for every v ∈ V .

[REF6] - 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 Graphs Chunk of text: Proposed data structure has construction time Oðn log2 n= log log nÞ and search time Oðn1Θð1= log log nÞ Þ in high dimensions and OðnαÞ; ð0oαo1Þin low dimensions. In general, currently there are no methods for effective exact NNS in high-dimensionality metric spaces. The reason behind this lies in the “curse” of dimensionality . To avoid the curse of dimensionality while retaining the logarithmic cost on the number of elements, it was proposed to reduce the requirements for the kNN problem solution, making it approximate (Approximate kNN). There are two commonly used definitions of the approximate neighbor search. One class of methods proposed to search with predefined accuracy ε (ε-NNS). It means that the distance between the query and any element in the result is no more than 1þε times the distance from query to its true k-th nearest neighbor.

[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: • How to pick a suitable set of parameters? • How does intrinsic dimensionality affect performance? The last question is answered by an empirical study with synthetic data. 4.1 Overall Performance Table 2 summarizes the performance of our method on all the datasets and similarity measures under two typical settings: the default setting (ρ = 1.0) achieving highest possible accuracy and a “fast” setting (ρ = 0.5) with slightly lower accuracy. We see that even with the fast setting, our method is able to achieve ≥ 95% recall, except for DBLP and Flickr. for which recall is below 90%. By putting in more computation with the default setting, we are able to boost recall for the more difficult datasets to close or above 90%.

[REF8] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Graph\_approaches/BIBREF72\_699a2e3b653c69aff5cf7a9923793b974f8ca164.pdf Title: Efficient and Robust Approximate Nearest Neighbor Search using Hierarchical Navigable Small World Graphs Chunk of text: However, the search on the layer always terminates before it reaches the element which belongs to the higher layer (otherwise the search on the upper layer would have stopped on a different element), so the probability of not reaching the target on s-th step is bounded by exp(-s· mL ). Thus the expected number of steps in a layer is bounded by a sum of geometric progression S =1/(1-exp(-mL )), which is independent of the dataset size. If we assume that the average degree of a node in the Delaunay graph is capped by a constant C in the limit of the large dataset (this is the case for random Euclid data , but can be in principle violated in exotic spaces), then the overall average number of distance evaluations in a layer is bounded by a constant C· S, independently of the dataset size. And since the expectation of the maximum layer index by the construction scales as O(log(N)), the overall complexity scaling is O(log(N)), in agreement with the simulations on low dimensional datasets. The inital assumption of having the exact Delaunay graph violates in Hierarchical NSW due to usage of approximate edge selection heuristic with a fixed number of neighbors per element. Thus, to avoid stucking into a local minimum the greedy search algorithm employs a backtracking procedure on the zero layer. Simulations show that at least for low dimensional data (Fig. 11, d=4) the dependence of the required ef parameter (which determines the complexity via the minimal number of hops during the backtracking) to get a fixed recall saturates with the rise of the dataset size.

[REF9] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Graph\_approaches/BIBREF71\_e2e073433931c4d1a739f548b7d17b6e9b2fa13e.pdf Title: The Small-World Phenomenon - An Algorithmic Perspective Chunk of text: The decentralized algorithm `4 is defined as follows: in each step, the current message-holder u chooses a contact that is as close to the target t as possible, in the sense of lattice distance. For j > 0, we say that the execution of .4 is in phase j when the lattice distance from the current node to t is greater than 2J and at most 2 j+l. We say .4 is in phase 0 when the lattice distance to t is at most 2. Thus, the initial value of j is at most log n. Now, because the distance from the message to the target decreases strictly in each step, each node that becomes the message holder has not touched the message before; thus, we may assume that the longrange contact from the message holder is generated at this moment. Suppose we are in phase j, log(10gn) <\_ j < logn, and the current message holder is u.

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

Title: Retrieval Architectures and Vector Search - Optimisations

In the field of neural information retrieval, retrieval architectures and vector search play a crucial role in optimizing the efficiency and effectiveness of the retrieval process. Various techniques and optimizations have been proposed to enhance the retrieval performance. This section discusses some of the key retrieval architectures and vector search optimizations that have been explored in the literature.

One approach to improve retrieval efficiency is through the use of encoding techniques such as Product Quantization (PQ) [REF0]. PQ encodings allow for the compression of vectors into smaller representations, which can significantly reduce the memory footprint and computational requirements during retrieval. Different PQ encodings, such as 16, 32, and 64 byte encodings, have been investigated to find the optimal trade-off between efficiency and quality [REF0].

Another important aspect of retrieval architectures is the selection of relevant passages or documents. Cross-attention models have been shown to be effective in reranking and selecting relevant passages [REF1]. These models leverage cross-attention between the question and the passage to improve the selection process. While cross-attention is not feasible for retrieving relevant passages in a large corpus, it has been successfully applied to selecting passages from a small number of retrieved candidates [REF1].

In the context of dense retrieval, several training methods have been explored to optimize the retrieval performance. Random negative sampling baselines, such as Rand Neg and In-Batch Neg, have been used to train dense retrieval models [REF2]. Rand Neg randomly samples negatives from the entire corpus, while In-Batch Neg uses other queries' relevant documents in the same batch as negative documents [REF2].

Efficiency and effectiveness are two key factors in retrieval systems. Query embedding pruning has been proposed as a technique to address the limitations of using all query embeddings for retrieval [REF3]. Not all query embeddings contribute equally to the effectiveness of the retrieval system, and many documents are retrieved by each query embedding. By selectively pruning query embeddings, it is possible to achieve effective and efficient dense retrieval [REF3].

Vector search optimizations also play a crucial role in improving retrieval efficiency. In the context of neural rankers, efficient computation of scores is essential. One approach is to compute a batch dot-product between the query and document embeddings, followed by reduction operations to obtain the scores of each document [REF4]. This computation is relatively cheap compared to existing neural rankers, as the cost is dominated by the gathering and transferring of pre-computed embeddings [REF4].

In addition to efficiency, retrieval effectiveness is also a key consideration. Constrained clustering has been shown to be effective in improving the retrieval effectiveness of dense retrieval models [REF5]. By applying constrained clustering, the memory and time efficiency of dense retrieval can be substantially improved, outperforming various retrieval models in terms of retrieval effectiveness, memory efficiency, and time efficiency [REF5].

While these retrieval architectures and vector search optimizations have shown promising results, further studies are required to fully understand their behavior and potential limitations [REF6]. Additionally, training efficiency is another important aspect to consider. End-to-end training has been shown to optimize the ranking performance for different compression techniques, making it suitable for improving the performance of compressed indexes [REF7].

In summary, retrieval architectures and vector search optimizations play a crucial role in enhancing the efficiency and effectiveness of neural information retrieval systems. Techniques such as encoding, passage selection models, query embedding pruning, and efficient computation of scores have been explored to improve retrieval performance. Further research is needed to fully understand the behavior and limitations of these techniques and to develop more efficient and effective retrieval architectures.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Optimisations/BIBREF73\_2cbb8de53759e75411bc528518947a3094fbce3a.pdf Title: Billion-Scale Similarity Search with GPUs Chunk of text: = 16, 32 and 64 byte PQ encodings for each vector. For Deep1B, we pre-process the vectors to d = 120 via OPQ, use |C1| = 218 and consider m = 20, 40. For a given encoding, we vary τ from 1 to 256, to obtain tradeoffs between efficiency and quality, as seen in Figure 5. 9Figure 6: Path in the k-NN graph of 95 million images from YFCC100M. The first and the last image are given; the algorithm computes the smoothest path between them. Discussion.

[REF1] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Optimisations/BIBREF50\_79cd9f77e5258f62c0e15d11534aea6393ef73fe.pdf Title: Dense Passage Retrieval for Open-Domain Question Answering Chunk of text: The passage selection model serves as a reranker through crossattention between the question and the passage. Although cross-attention is not feasible for retrieving relevant passages in a large corpus due to its nondecomposable nature, it has more capacity than the dual-encoder model sim(q, p) as in Eq. (1). Applying it to selecting the passage from a small number of retrieved candidates has been shown to work well (Wang et al., 2019, 2018; Lin et al., 2018). Specifically, let Pi ∈ R L×h (1 ≤ i ≤ k) be a BERT (base, uncased in our experiments) representation for the i-th passage, where L is the maximum length of the passage and h the hidden dimension. The probabilities of a token being the starting/ending positions of an answer span and a passage being selected are defined as: Pstart,i(s) = softmax Piwstart s , (3) Pend,i(t) = softmax Piwend t , (4) Pselected(i) = softmax Pˆ|wselected

[REF2] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Optimisations/BIBREF52\_7b577ba0e4230b2ac58d297b3d2cfc3d2f1aaace.pdf Title: Optimizing Dense Retrieval Model Training with Hard Negatives Chunk of text: . 7.2.2 Dense Retrieval. The DR baselines include several popular training methods. For random negative sampling baselines, we present Rand Neg and In-Batch Neg [15, 33]. The former randomly samples negatives from the entire corpus, and the latter uses other queries’ relevant documents in the same batch as negative documents.

[REF3] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Optimisations/BIBREF77\_c3cf35677834fb535d3bc7cf8d375366df4b1397.pdf Title: Query Embedding Pruning for Dense Retrieval Chunk of text: In short, our analysis shows that not all query embeddings are needed for effective retrieval – indeed, this implies that some query embeddings can be ignored without negatively impacting the effectiveness of the whole system. Moreover, many documents are retrieved by each query embedding – the ramification is that all encoded document embeddings must be stored in memory, as many documents must be scored by the exact ranker to obtain effective results. Finally, less new documents are retrieved by later query (masked) embeddings, as these embeddings are reduntant w.r.t. to earlier embeddings. In order to address the aforementioned limitations, we aim to use less query embeddings for effective and efficient dense retrieval. Indeed, we desire high effectiveness with less query embeddings. In the next section we discuss how dynamic pruning can be exploited in dense retrieval to increase efficiency without negatively impacting effectiveness. 3 QUERY EMBEDDING PRUNING As we have shown in Section 2, the query embeddings do not contribute equally to the final effectiveness of the retrieved document set.

[REF4] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Optimisations/BIBREF55\_60b8ad6177230ad5402af409a6edb5af441baeb4.pdf Title: ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT Chunk of text: We pad the k documents to their maximum length to facilitate batched operations, and move the tensor D to the GPU’s memory. On the GPU, we compute a batch dot-product of Eq and D, possibly over multiple mini-batches. e output materializes a 3-dimensional tensor that is a collection of cross-match matrices between q and each document. To compute the score of each document, we reduce its matrix across document terms via a max-pool (i.e., representing an exhaustive implementation of our MaxSim computation) and reduce across query terms via a summation. Finally, we sort the k documents by their total scores. Relative to existing neural rankers (especially, but not exclusively, BERT-based ones), this computation is very cheap that, in fact, its cost is dominated by the cost of gathering and transferring the pre-computed embeddings. To illustrate, ranking k documents via typical BERT rankers requires feeding BERT k dierent inputs each of length l = |q| + |di | for query q and documents di , where aention has quadratic cost in the length of the sequence.

[REF5] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Optimisations/BIBREF76\_91429255eefe48ad140ccfaf6aa1e6be11a72a53.pdf Title: Learning Discrete Representations via Constrained Clustering for Effective and Efficient Dense Retrieval Chunk of text: We conduct experiments on two widely-adopted ad-hoc retrieval benchmarks. Experimental results show that RepCONC significantly outperforms competitive quantization baselines andLearning Discrete Representations via Constrained Clustering for Effective and Efficient Dense Retrieval WSDM’22, February 21-25, 2022, Phoenix, Arizona substantially improves the memory efficiency and time efficiency of DR. It substantially outperforms various retrieval models in terms of retrieval effectiveness, memory efficiency, and time efficiency. The ablation study demonstrates that constrained clustering is the key to the effectiveness of RepCONC.

[REF6] - 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: We include additional experiments in Appendix. A.2 discusses the surprisingly small overlap (<25%) between dense retrieval results and sparse retrieval results. DR is a fundamentally different approach and more studies are required to understand its behavior. A.3 and A.4 study the asynchronous gaps and hyperparameters. A.5 includes case studies that the irrelevant documents from ANCE are often still “semantically related” and very different from those made by sparse retrieval. 7Preprint 0k 5k 10k15k20k25k30k

[REF7] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Optimisations/BIBREF52\_7b577ba0e4230b2ac58d297b3d2cfc3d2f1aaace.pdf Title: Optimizing Dense Retrieval Model Training with Hard Negatives Chunk of text: We can see that end-to-end training better optimizes the ranking performance for different compression techniques. Thus, ADORE is suitable to improve the performance of compressed indexes. 8.4 Training Efficiency This section presents the training efficiency of our proposed methods to answer RQ3 from two aspects, namely training time and computational resources. Since ANCE is competitive in terms of effectiveness, we use it as our efficiency baseline. 8.4.1 Training Time. We test the training speed with 11GB GeForce RTX 2080 Ti GPUs and show the results in Table 4.

[REF8] - paperID: ./papers\_pdf/paper\_section/Retrieval\_Architectures\_and\_Vector\_Search-Optimisations/BIBREF75\_d6f83c915565f575e55fdce0424f65fe192af218.pdf Title: Jointly Optimizing Query Encoder and Product Quantization to Improve Retrieval Performance Chunk of text: Model Index MARCO Passage DL Passage Index MARCO Doc DL Doc GB MRR@10 R@100 NDCG@10 R@100 GB MRR@100 R@100 NDCG@10 Traditional BoW BM25 0.59 0.187\*\* 0.670\*\* 0.497\*\* 0.460 2.17 0.278\*\* 0.807\*\* 0.523\*\* Augmented BoW doc2query 0.65 0.215\*\* 0.713\*\* 0.533 \*\* 0.471 n.a. n.a. n.a. n.a.

[REF9] - 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: This fundamentally aligns the distribution of negative samples in training and of irrelevant documents to separate in testing. From the variance reduction point of view, these ANCE negatives lift the upper bound of per instance gradient norm, reduce the variance of the stochastic gradient estimation, and lead to faster learning convergence. We implement ANCE using an asynchronously updated ANN index of the corpus representation. Similar to Guu et al. (2020), we maintain an Inferencer that parallelly computes the document encodings with a recent checkpoint from the being optimized DR model, and refresh the ANN index used for negative sampling once it finishes, to keep up with the model training. Our experiments demonstrate the advantage of ANCE in three text retrieval scenarios: standard web search (Craswell et al., 2020), OpenQA (Rajpurkar et al., 2016; Kwiatkowski et al., 2019), and in a commercial search engine’s retrieval system. We also empirically validate our theory that the gradient norms on ANCE sampled negatives are much bigger than local negatives and thus improve the convergence of dense retrieval models.

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

Title: Learned Sparse Retrieval - Document expansion learning

In the context of neural information retrieval, learned sparse retrieval techniques have gained significant attention due to their ability to improve retrieval performance by incorporating learned representations and models. One approach in this area is document expansion learning, which aims to enhance the retrieval process by expanding the original document representation with additional information.

Gradual unfreezing is a fine-tuning method that has been applied to language models to improve their performance [REF0]. This approach involves gradually unfreezing the layers of the model during fine-tuning, starting from the top layer and progressively including lower layers. By updating the parameters of the model's layers over time, gradual unfreezing allows for better adaptation and optimization. In the context of encoder-decoder models, layers in both the encoder and decoder are unfrozen in parallel, starting from the top. This approach ensures that the shared parameters, such as the input embedding matrix and output classification matrix, are updated throughout the fine-tuning process.

Support vector regression (SVR) with the radial basis function (RBF) kernel has been used in the prediction of term weights for retrieval models [REF1]. SVR maps the original features into a higher dimensional space and aims to find a linear solution that is as flat as possible in this space. The RBF kernel, in particular, measures vector similarity based on the Gaussian distribution function, allowing for the localization of the impact of training samples during testing. This enables SVR to fit complex non-linear regression models. In the context of retrieval, SVR with the RBF kernel has shown promising results in improving term weight prediction.

When conducting experiments and studies in neural information retrieval, it is important to consider the trade-off between exploring various factors and keeping the study feasible [REF2]. While a comprehensive exploration of all factors may be prohibitively expensive, it is still valuable to consider combinations of different approaches. For instance, encoder-only models like BERT are designed for specific tasks such as classification or span prediction, but may not be suitable for generative tasks like translation or summarization. Therefore, it is necessary to adapt existing approaches and architectures to suit the specific requirements of the study.

Efficient term recall prediction can significantly speed up the retrieval process by reusing previously determined recall values [REF3]. By avoiding the generation of features and prediction stages, the recall value of a previous occurrence of the same term can be reused. This approach leverages the regularity of recall values for recurring terms, leading to improved efficiency in retrieval. Additionally, the prediction framework for recall values in retrieval can be seen as a transfer learning task, where learning occurs across different queries and contributes to the overall retrieval performance.

Term mismatch is a well-known challenge in retrieval models, and various methods have been proposed to address this issue [REF4]. These methods aim to improve the matching between terms in documents and queries, either by considering the document's end, the query's end, or both ends. Semantic analysis techniques, such as concept identification or synonym identification, have also been employed to enhance the semantic level matching in retrieval. Furthermore, diagnostic interventions have been explored to improve problematic areas of queries, leading to more effective retrieval models.

The architecture of encoder-decoder models with shared parameters and full attention across input and target sequences has shown similarities to BERT for classification tasks [REF5]. By following a text-to-text framework, the prefix language model (LM) architecture closely resembles BERT. This architecture has been applied to various tasks, such as natural language inference, where the premise and hypothesis are transformed into a sequence for language modeling. The utilization of encoder-decoder models in retrieval tasks provides a flexible and adaptable approach to address different information retrieval challenges.

Sparse representation learning methods, such as SparTerm, offer opportunities to improve the ranking performance of term-based representations while maintaining interpretability and efficiency [REF6]. SparTerm has demonstrated superior performance compared to previous sparse models, even outperforming models based on larger pre-trained language models. The transfer of deep knowledge from pre-trained language models to sparse methods has provided valuable insights into sparse representation learning. This research area intersects with bag-of-words (BoW) methods and pre-trained language models for text retrieval, offering new avenues for exploration and improvement.

In the context of recall-based term weighting, predicted recall values have been utilized as user term weights in retrieval experiments [REF7]. Various prediction features, such as inverse document frequency (idf), term centrality, concept centrality, replaceability, and abstractness, have been employed to estimate recall values. These models have been trained on previous TREC queries and tested using cross-validation on subsequent TREC queries. The performance of the models has been evaluated using different cross-validation techniques, demonstrating the effectiveness of recall-based term weighting.

The theoretical consistency between multiple pocket document models and single pocket relevance models remains an open question for future research [REF8]. While document expansion learning techniques have shown promising results, further investigation is needed to understand the theoretical implications and potential limitations of these models. Experimental studies have been conducted to evaluate the performance of retrieval using recall-based term weighting, comparing it with baselines such as language models with Dirichlet smoothing and Okapi BM25. These studies provide insights into the effectiveness of recall term weighting and its impact on retrieval performance.

The size of the dataset plays a crucial role in the overfitting of models to specific tasks [REF9]. To address this issue, mixing proportions can be set based on the size of each task's dataset. However, it is important to consider the inclusion of unsupervised tasks, which may have significantly larger datasets compared to supervised tasks. Sampling in proportion to each dataset's size may result in undertraining on supervised tasks. Therefore, careful consideration is required to ensure a balanced training process that accounts for the varying sizes of datasets across different tasks.

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Document\_expansion\_learning/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: We experiment with various values for d. The second alternative fine-tuning method we consider is “gradual unfreezing” (Howard and Ruder, 2018). In gradual unfreezing, more and more of the model’s parameters are finetuned over time. Gradual unfreezing was originally applied to a language model architecture consisting of a single stack of layers. In this setting, at the start of fine-tuning only the parameters of the final layer are updated, then after training for a certain number of updates the parameters of the second-to-last layer are also included, and so on until the entire network’s parameters are being fine-tuned. To adapt this approach to our encoder-decoder model, we gradually unfreeze layers in the encoder and decoder in parallel, starting from the top in both cases. Since the parameters of our input embedding matrix and output classification matrix are shared, we update them throughout fine-tuning.

[REF1] - 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: It maps the original features into a higher dimensional space, and looks for a linear solution that is as flat as possible in the higher dimension space. With a proper kernel, the trained regression model can be non-linear. In our pilot study we tested support vector regression with linear, polynomial and RBF kernels using SVM-light version 6.023 . Results show that the RBF kernel performs the best. The RBF kernel measures vector similarity according to the RBF function (or the Gaussian distribution function), where the vector similarity drops exponentially with the squared distance between the vectors. Because the RBF kernel effectively localizes the impact of the training samples during testing, RBF support vector regression can fit very complex non-linear regression models. Except simply scaling the features which happens during preprocessing, the final regression model treats all features the same.

[REF2] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Document\_expansion\_learning/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: This “coordinate ascent” approach might miss second-order effects (for example, some particular unsupervised objective may work best on a model larger than our baseline setting), but performing a combinatorial exploration of all of the factors in our study would be prohibitively expensive. In future work, we expect it could be fruitful to more thoroughly consider combinations of the approaches we study. Our goal is to compare a variety of different approaches on a diverse set of tasks while keeping as many factors fixed as possible. In order to satisfy this aim, in some cases we do not exactly replicate existing approaches. For example, “encoder-only” models like BERT (Devlin et al., 2018) are designed to produce a single prediction per input token or a single prediction for an entire input sequence. This makes them applicable for classification or span prediction tasks but not for generative tasks like translation or abstractive summarization. As such, none of the model architectures we consider are identical to BERT or consist of an encoder-only structure.

[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: Recall values do not vary more than 0.2, for most (two thirds) of the occurrences of recurring terms. This regularity can be utilized to speed up recall prediction. Whenever we are certain that the recall value of a previous occurrence of the same term can be reused, we can avoid all the feature generation and prediction stages altogether. Efficient term recall prediction is the focus of Chapter 7. 5.4 Discussion – Transcendental Features and Retrieval Modeling as a Transfer Learning Task This regression prediction framework for predicting P(t|R) is just a simple and straightforward application of regression learning from statistics. However, a significantly new idea here is that the learning happens across the different queries, which is consistent with the transfer learning formalism. Traditionally, the retrieval of collection documents according to a given query is treated as one document classification task, where each document needs to be classified as relevant or non-relevant to the query.

[REF4] - 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: More generally, term weight prediction methods are also related because of their use of predicted term weights to improve retrieval, and are reviewed in Section 2.4. It is widely accepted that term mismatch is an important problem in retrieval. Even though there has been no clear understanding of what exact role term mismatch plays in the retrieval models and the retrieval process, a plethora of methods were proposed to solve mismatch. They worked from the document’s end, the query’s end, or both ends, and are reviewed in Section 2.6. Semantic analysis of texts or queries (Section 2.7) is a standard technique to improve semantic level matching in retrieval. Examples include concept identification in queries or synonym identification for query terms. Another aspect of this research is diagnostic interventions that improve problem areas of a query.

[REF5] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Document\_expansion\_learning/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: This architecture is similar to an encoder-decoder model with parameters shared across the encoder and decoder and with the encoder-decoder attention replaced with full attention across the input and target sequence. We note that when following our text-to-text framework, the prefix LM architecture closely resembles BERT (Devlin et al., 2018) for classification tasks. To see why, consider an example from the MNLI benchmark where the premise is “I hate pigeons. ”, the hypothesis is “My feelings towards pigeons are filled with animosity.” and the correct label is “entailment”. To feed this example into a language model, we would transform it into the sequence “mnli premise: I hate pigeons. hypothesis: My feelings towards pigeons are filled with animosity.

[REF6] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Document\_expansion\_learning/BIBREF87\_57a07372e2a620d6ae920f74877eee5f61753a96.pdf Title: SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval Chunk of text: The proposed SparTerm indicates that there is much space for improving the ranking performance of termed-based representations, while still keeping the interpretability and efficiency of BoW methods. Evaluated on MSMARCO dataset, SparTerm significantly outperforms previous sparse models based on the comparable size of PLMs. The top-ranking performance of SparTerm even outperforms Doc2Query-T5, which is based on the pre-trained model of 2x model size and 70x pre-training corpus size. Moreover, we conduct further empirical analysis about how the deep knowledge of PLMs can be transferred to the sparse method, which gives new insights for sparse representation learning. 2 RELATED WORK Our work relates to two research fields: bag-of-words representations and pre-trained language model for text retrieval. 2.1 Bag-of-words Methods Bag-of-words(BoW) methods have played a central role in the firststage retrieval. These methods convert a document or query into a set of single terms, and each term associates a weight to characterize its weight.

[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: In Table 6.5, we present results using predicted recall values (Chapter 5) as user term weights. Prediction features used here are idf, term centrality, concept centrality, replaceability and abstractness. Models were trained on TREC queries from previous year(s), and were tested using 5-fold cross validation on the 50 TREC queries of the next year. This means, the RBF support vector regression model was always trained on the 50 training queries (if training set includes only one TREC dataset). 50 test queries were split into 5 folds, 4 of which were used as development set to tune meta-parameters, 1 fold was used for testing. However, the learning model does not require that much development data. 2-fold cross validation uses fewer (only 25) development queries, and still yields the same performance and optimal parameter values as 5-fold cross validation does.

[REF8] - 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: Overall, the multiple pocket document models are not theoretically consistent with the single pocket relevance models, which demands a cleaner explanation. We point out this problem and leave it as future work. 726.2 Experiments – Retrieval Using 2-pass P(t|R) Prediction This section presents retrieval experiments using recall based term weighting for both estimated true recall values from relevance judgments and predicted recall based on supervised learning. Ablation studies are presented to provide a sense of what features are most effective. The basic baselines are the language model with Dirichlet smoothing and Okapi BM25. However, other baselines such as Relevance Model are also included to provide a better understanding of how and why recall term weighting works.

[REF9] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Document\_expansion\_learning/BIBREF36\_3cfb319689f06bf04c2e28399361f414ca32c4b3.pdf Title: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer Chunk of text: A similar exploration was performed by Wang et al. (2019a). Examples-proportional mixing A major factor in how quickly a model will overfit to a given task is the task’s data set size. As such, a natural way to set the mixing proportions is to sample in proportion to the size of each task’s data set. This is equivalent to concatenating the data sets for all tasks and randomly sampling examples from the combined data set. Note, however, that we are including our unsupervised denoising task, which uses a data set that is orders of magnitude larger than every other task’s. It follows that if we simply sample in proportion to each data set’s size, the vast majority of the data the model sees will be unlabeled, and it will undertrain on all of the supervised tasks. Even without the unsupervised task, some tasks (e.g. WMT English to French) are so large that they would similarly crowd out most of the batches.

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

Title: Learned Sparse Retrieval - Impact score learning

Impact score learning is a technique used in neural information retrieval to assign semantic importance to each token of a document without the need for queries [REF2]. This approach leverages the learned term-weighting scheme to predict the relevance of each token and generate impact scores for efficient retrieval [REF2]. The impact scores are computed by representing each document as a list of term-score pairs, which are then converted into an inverted index [REF2]. This index can be deployed and searched as usual for efficient query processing [REF2].

One important aspect of impact score learning is the computation of impact scores. To address the space requirements of storing floating-point values per posting, quantized impact scores are used, which belong to the range of [1, 2^b - 1], where b is the number of bits used to store each value [REF5]. Experimental results have shown that quantizing the scores using linear quantization with b = 8 does not result in any noticeable loss in precision compared to the original scores [REF5]. During query processing, the quantized scores of the document terms matching the query are summed up to compute the query-document score [REF5].

DeepImpact is a neural model that utilizes impact score learning for efficient retrieval [REF2]. It has been compared with other methods such as BM25, DeepCT, and DocT5Query, and has shown promising results as a first-stage ranker [REF7]. However, it has been observed that the distribution of scores induced by BM25, which is exploited more efficiently by the MaxScore algorithm, is not efficiently exploited by DeepImpact due to its learned scores [REF3]. Further research is needed to optimize the query processing speed of DeepImpact [REF3].

In summary, impact score learning is a technique in neural information retrieval that assigns semantic importance to each token of a document without the need for queries. It leverages a learned term-weighting scheme to predict the relevance of each token and generate impact scores. These scores are quantized and stored in an inverted index for efficient query processing. DeepImpact is an example of a neural model that utilizes impact score learning for efficient retrieval, although further research is needed to optimize its query processing speed [REF2][REF3][REF5][REF7].

References given to GPT:

[REF0] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Impact\_score\_learning/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: r reduction, used as a performance measure. d a document, as indexed by an IR system. q a query, as processed by an IR system, i.e., a set of terms. N the number of documents indexed by the IR system. t, ti a term, as may exist within a query. Scoreq(d) a generic query-document ranking function. st(q, d) a generic term-document similarity function.

[REF1] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Impact\_score\_learning/BIBREF91\_4aa1d28944856ebe1950a27f633c6667ead3cbf8.pdf Title: Learning Passage Impacts for Inverted Indexes Chunk of text: However, several recent studies [7, 14] have shown that this can have very high computational cost, even if re-ranking just the top 1000 results. Other studies [9, 12, 13] proposed methods with lower lower computational cost but typically some loss in retrieval quality. BERT’s Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

[REF2] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Impact\_score\_learning/BIBREF91\_4aa1d28944856ebe1950a27f633c6667ead3cbf8.pdf Title: Learning Passage Impacts for Inverted Indexes Chunk of text: Max input text length was set to 160 tokens. Losses are back-propagated through the whole DeepImpact neural model with a learning rate of 3×10−6 with the Adam optimizer. We used batches of 32 triples and train for 100,000 iterations. Impact Scores Computation. Following the training phase, DeepImpact can leverage the learned term-weighting scheme to predict the semantic importance of each token of the documents without the need for queries. Each document is represented as a list of term-score pairs, which are converted into an inverted index. The index can then be deployed and searched as usual for efficient query processing.

[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: We also see that DeepImpact mean response time exceeds the time reported for other methods. We trace this to the query processing strategy: the distribution of scores induced by BM25, used in BM25, DeepCT, and DocT5Query is exploited more efficiently by 2https://github.com/jmmackenzie/term-weighting-efficiency 3https://github.com/castorini/docTTTTTquery 4https://github.com/Georgetown-IR-Lab/epic-neural-ir 5https://github.com/stanford-futuredata/ColBERT 6https://github.com/DI4IR/SIGIR2021the MaxScore algorithm. In contrast, DeepImpact learns new scores, whose distribution is not efficiently exploited by MaxScore. We performed additional experiments using disjunctive query processing without optimizations, omitted for space limitations. These experiments show DeepImpact to be in line with the speed of the other approaches. Optimizing the query processing speed of DeepImpact is an interesting open problem for future research. Table 2: Effectiveness metrics and mean response time (MRT, in ms) for first-stage methods, on MSMARCO Dev Queries, TREC 2019 queries, and TREC 2020 queries.

[REF4] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Impact\_score\_learning/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: Figure 1.3 provides the main infrastructure that is discussed in this survey. We will focus on the “online” components, e.g., those responsible for the cascading components of search, while referring to the “offline” components whenever it is necessary. The remainder of this survey is structured as follows: • Chapter 2 provides an overview of the modern infrastructure foundations within a search engine, covering the basic form of the inverted index data structure, and the essentials of query processing. • Chapter 3 provides an introduction to approaches for increasing the efficiency of query processing, namely the dynamic pruning techniques. • Chapter 4 describes query efficiency predictors – a new technique to estimate the response time of queries – that is gaining attention Full text available at: <http://dx.doi.org/10.1561/150000005711> for a number of applications involving efficient retrieval on a per-query basis. • Chapter 5 provides an overview of impact-sorted indexes, which make offline changes to the layout of the inverted index in order to improve the efficiency of query processing. • Chapter 6 provides an overview of cascading search architectures, and provides insights into how to efficiently deploy learning-torank, a retrieval technique known to benefit the search engine’s effectiveness by re-ranking a set of K documents.

[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: Since storing a floating point value per posting would blow up the space requirements of the inverted index, we decided to store impacts in a quantized form. The quantized impact scores belong to the range of [1, 2 𝑏 − 1], where 𝑏 is the number of bits used to store each value. We experimented with 𝑏 = 8 using linear quantization, and did not notice any loss in precision w.r.t. the original scores. Since we quantized all the scores in the index in the same way, to compute a query-document score at query processing we can just sum up all the quantized scores of the document terms matching the query. 3 EXPERIMENTAL RESULTS In this section, we analyze the performance of the proposed method with an extensive experimental evaluation in a realistic and reproducible setting, using state-of-the-art baselines and a standard test collection and query logs. Hardware.

[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: 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. Baselines: Experiments were done with done with three baseline and three experimental indexing methods, as described below. tf index is a standard tf -based index, e.g., as used by BM25. TextRank is a widely-used unsupervised graph-based term weighting approach. We use the open source PyTextRank implementation4 .

[REF7] - 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 initial 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 proposed 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.

[REF8] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Impact\_score\_learning/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: “Reducing Query Latencies in Web Search Using Fine-Grained Parallelism”. World Wide Web. 12(4): 441. issn: 1573- 1413. doi: 10.1007/s11280-009-0066-4. url: <https://doi.org/10>. 1007/s11280-009-0066-4. Freire, A., C. Macdonald, N. Tonellotto, I. Ounis, and F. Cacheda. 2012. “Scheduling Queries Across Replicas”.

[REF9] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Impact\_score\_learning/BIBREF3\_629f50daebbb9003f645f671f76cc6b33088c17d.pdf Title: Efficient Query Processing for Scalable Web Search Chunk of text: 81 4 Query Efficiency Prediction for Dynamic Pruning 82 4.1 Implementations of Query Efficiency Prediction . . . . . . 84 4.2 Delayed Query Efficiency Prediction . . . . . . . . . . . . 86 4.3 Query Efficiency Prediction Applications . . . . . . . . . . 87 4.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 96 Full text available at: <http://dx.doi.org/10.1561/15000000575> Impact-Sorted Indexes 98 5.1 Data Structures . . . . . . . . . . . . . . . . . . . . . . . 100 5.2 Query Processing . . . . . . . . . . . . . . . . . . . . . .

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

Title: Learned Sparse Retrieval - Sparse representation learning

Sparse representation learning has gained significant attention in the field of Neural Information Retrieval (NIR) due to its ability to capture important features while reducing the dimensionality of the data. This section focuses on the use of learned sparse retrieval techniques, specifically sparse representation learning, in the context of NIR.

One popular approach in learned sparse retrieval is the use of dense retrieval models based on BERT Siamese architectures [REF0]. These models have become the standard approach for candidate generation in Question Answering and Information Retrieval tasks. Recent works have highlighted the importance of training strategies to achieve state-of-the-art results, including improved negative sampling and distillation techniques [REF0]. However, these models often suffer from scalability issues when applied to large collections, as they require storing embeddings for each (sub)term [REF0].

To address the scalability concern, approximate nearest neighbors (ANN) search has been proposed as a solution [REF0]. ANN search allows for efficient retrieval by finding approximate nearest neighbors instead of exact matches. However, the impact of using ANN search on IR metrics has been relatively unexplored [REF0]. Most studies report results using exact, brute-force search, providing limited insights into the actual computational cost of these models [REF0].

Sparse lexical representations have been proposed as an effective approach for first-stage ranking in learned sparse retrieval [REF1]. The SparTerm model predicts the importance of each token in a BERT WordPiece vocabulary based on the logits of the Masked Language Model (MLM) layer [REF1]. The importance predictors are then summed over the input sequence tokens to obtain the final representation [REF1]. By applying ReLU, the positivity of term weights is ensured [REF1]. This approach allows for efficient and effective ranking, as demonstrated by the SPLADE model [REF1].

In terms of efficiency, pruning techniques have been explored to reduce the dimensionality of sparse representations [REF2]. By pruning the vectors, the vocabulary size can be significantly reduced without a significant impact on ranking effectiveness [REF2]. For example, in one study, pruning to a vocabulary size of 1000 resulted in virtually no difference in ranking effectiveness compared to the original vocabulary size [REF2]. Additionally, the computation of query representations in sparse retrieval models can be parallelized with the initial retrieval process, minimizing the impact on query-time latency [REF2].

Joint training of importance predictors and gating controllers has also been investigated in learned sparse retrieval [REF3]. By leveraging the supervisory ranking signal, the training of these components can be guided to improve overall performance [REF3].

Experimental evaluations of learned sparse retrieval techniques have been conducted on benchmark datasets such as MSMARCO [REF3]. These datasets consist of real-world queries and passages, allowing for comprehensive evaluation of the proposed models [REF3].

In conclusion, learned sparse retrieval techniques, particularly sparse representation learning, have shown promise in improving the efficiency and effectiveness of information retrieval systems. By leveraging techniques such as approximate nearest neighbors search, sparse lexical representations, pruning, and joint training, researchers have made significant advancements in this field. However, further research is needed to explore the scalability and generalizability of these techniques to larger collections and diverse domains.

References given to GPT:

[REF0] - 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: 2 RELATED WORKS Dense retrieval based on BERT Siamese models has become the standard approach for candidate generation in Question Answering and IR [8, 10, 12, 15, 25]. While the backbone of these models remains the same, recent works highlight the critical aspects of the training strategy to obtain state-of-the-art results, ranging from improved negative sampling [8, 25] to distillation [11, 15]. ColBERT pushes things further: the postponed token-level Short Research Paper III SIGIR ’21, July 11–15, 2021, Virtual Event, Canada 2288interactions allow to efficiently apply the model for first-stage retrieval, benefiting of the effectiveness of modeling fine-grained interactions, at the cost of storing embeddings for each (sub)term – raising concerns about the actual scalability of the approach for large collections. To the best of our knowledge, very few studies have discussed the impact of using approximate nearest neighbors (ANN) search on IR metrics [2, 23]. Due to the moderate size of the MS MARCO collection, results are usually reported with an exact, brute-force search, therefore giving no indication on the effective computing cost.

[REF1] - 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: Our SPLADE model relies on such regularization, as well as other key changes, that boost both the efficiency and the effectiveness of this type of models. 3 SPARSE LEXICAL REPRESENTATIONS FOR FIRST-STAGE RANKING In this section, we first describe in details the SparTerm model , before presenting our model named SPLADE. 3.1 SparTerm SparTerm predicts term importance – in BERT WordPiece vocabulary (|𝑉 | = 30522) – based on the logits of the Masked Language Model (MLM) layer. More precisely, let us consider an input query or document sequence (after WordPiece tokenization) 𝑡 = (𝑡1, 𝑡2, ..., 𝑡𝑁 ), and its corresponding BERT embeddings (ℎ1, ℎ2, ..., ℎ𝑁 ). We consider the importance 𝑤𝑖𝑗 of the token 𝑗 (vocabulary) for a token 𝑖 (of the input sequence): 𝑤𝑖𝑗 = transform(ℎ𝑖) 𝑇 𝐸𝑗 + 𝑏𝑗 𝑗 ∈ {1, ..., |𝑉 |} (1) where 𝐸𝑗 denotes the BERT input embedding for token 𝑗, 𝑏𝑗 is a token-level bias, and transform(.) is a linear layer with GeLU activation and LayerNorm. Note that Eq. 1 is equivalent to the MLM prediction, thus it can be also be initialized from a pre-trained MLM model. The final representation is then obtained by summing importance predictors over the input sequence tokens, after applying ReLU to ensure the positivity of term weights: 𝑤𝑗 = 𝑔𝑗 ×

[REF2] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Sparse\_representation\_learning/BIBREF94\_0c57dcf959ead9530f9ec3ebe0dd58de42a3e8af.pdf Title: Expansion via Prediction of Importance with Contextualization Chunk of text: We show the effectiveness and efficiency of r = 2000 (reduces vocabulary by 93.4%) and r = 1000 (96.7%) in Table 1. We observe that the vectors can be pruned to r = 1000 with virtually no difference in ranking effectiveness (differences not statistically significant). We also tested with lower values of r, but found that the effectiveness drops off considerably by r = 100 (0.241 and 0.285 for BM25 and docTTTTTquery, respectively). Ranking efficiency. We find that EPIC can be implemented with a minimal impact on query-time latency. On average, the computation of the query representation takes 18ms on GPU and 51ms on CPU. Since this initial stage retrieval does not use our query representation, it is computed in parallel with the initial retrieval, which reduces the impact on latency.

[REF3] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Sparse\_representation\_learning/BIBREF87\_57a07372e2a620d6ae920f74877eee5f61753a96.pdf Title: SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval Chunk of text: In the experiment, we set 𝜆2 much larger than 𝜆1 to encourage more terms to be expanded. End-to-end joint training. Intuitively, the supervisory ranking signal can also be leveraged to guide the training of the gating controller, thus we can train the importance predictor and gating controller jointly: 𝐿 = 𝐿𝑟𝑎𝑛𝑘 + 𝐿𝑒𝑥𝑝 (9) 4 EXPERIMENTAL SETUP 4.1 Datasets and Metrics We evaluate our method on MSMARCO which consists of two benchmark datasets: MSMARCO Passage Retrieval dataset is based on the public MSMARCO dataset with a collection of 8.8M passages from Web pages gathered from Bing’s results to 1M real-world queries. Each query is associated with one or very few passages marked as relevant while no passage explicitly indicated as irrelevant.

[REF4] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Sparse\_representation\_learning/BIBREF96\_9f753f67da834e59f9a5c8cdf9a88ee84c496b2d.pdf Title: Minimizing FLOPS to Learn Efficient Sparse Representations Chunk of text: The FLOPs F(fθ,P) being a discontinuous function of model parameters, is hard to optimize, and hence we will instead optimize using a continuous relaxation of it. Denote by `(fθ, D), any metric loss on D for the embedding function fθ. The goal in this paper is to minimize the loss while controlling the expected FLOPs F(fθ,P) defined in Eqn. 2. Since the distribution P is unknown, we use the samples to get an estimate of F(fθ,P). Recall the empirical fraction of non-zero activations p¯j = 1 n Pn i=1 I[fθ(xi)j 6= 0], which converges in probability to pj . Therefore, with a slight abuse of notation define F(fθ, D) = Pd j=1 p¯ 2 j , which is a consistent estimator for F(fθ,P) based on the samples D. Note that F denotes either the population or empirical quantities depending on whether the functional argument is P or D. We now consider the following regularized loss. min θ∈Θ `(fθ, D) +

[REF5] - 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: We include a pure lexical SparTerm trained with our ranking pipeline (ST lexical-only). To illustrate the benefits of the log-saturation, we add results for models trained using Eq. (2) instead of Eq. (4) (ST exp-ℓ1 and ST exp-ℓFLOPS). For sparse models, we indicate an estimation of the average number of floating-point operations between a query and a document in Table 1, when available, which is defined as the expectation E𝑞,𝑑 hÍ 𝑗 ∈𝑉 𝑝 (𝑞) 𝑗 𝑝 (𝑑) 𝑗 i where 𝑝𝑗 is the activation probability for token 𝑗 in a document 𝑑 or a query 𝑞. It is empirically estimated from a set of approximately 100k development queries, on the MS MARCO collection.

[REF6] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Sparse\_representation\_learning/BIBREF94\_0c57dcf959ead9530f9ec3ebe0dd58de42a3e8af.pdf Title: Expansion via Prediction of Importance with Contextualization Chunk of text: These terms are then indexed and used for retrieval using BM25. The docTTTTTquery model uses a pre-trained T5 model . - Duet , a hybrid representation- and interaction-focused model. We include the top Duet variant on the MS-MARCO leaderboard (version 2, ensemble) to compare with another model that utilizes query and document representations. - TK , a contextualized interaction-based model, focused on minimizing query time.

[REF7] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Sparse\_representation\_learning/BIBREF96\_9f753f67da834e59f9a5c8cdf9a88ee84c496b2d.pdf Title: Minimizing FLOPS to Learn Efficient Sparse Representations Chunk of text: Exact retrieval of the top-k nearest neighbours is expensive in practice for high-dimensional dense embeddings learned from deep neural networks, with practitioners often resorting to approximate nearest neighbours (ANN) for efficient retrieval. Popular approaches for ANN include Locality sensitive hashing (LSH) (Gionis et al., 1999; Andoni et al., 2015; Raginsky and Lazebnik, 2009) relying on random projections, Navigable small world graphs (NSW) (Malkov et al., 2014) and hierarchical NSW (HNSW) (Malkov and Yashunin, 2018) based on constructing efficient search graphs by finding clusters in the data, Product Quantization (PQ) (Ge et al., 2013; Jegou et al., 2011) approaches which decompose the original space into a cartesian product of low-dimensional subspaces and quantize each of them separately, and Spectral hashing (Weiss et al., 2009) which involves an NP hard problem of computing an optimal binary hash, which is relaxed to continuous valued hashes, admitting a simple solution in terms of the spectrum of the similarity matrix. Overall, for compact representations and to speed up query times, most of these approaches use a variety of carefully chosen data structures, such as hashes (Neyshabur and Srebro, 2015; Wang et al., 2018), locality sensitive hashes (Andoni et al., 2015), inverted file structure (Jegou et al., 2011; Baranchuk et al., 2018), trees (Ram and Gray, 2012), clustering (Auvolat et al., 2015), quantization sketches (Jegou et al., 2011; Ning et al., 2016), as well as dimensionality reductions based on principal component analysis and t-SNE (Maaten and Hinton, 2008). End to end ANN. Learning the ANN structure end-to-end is another thread of work that has gained popularity recently. Norouzi et al. (2012) propose to learn binary representations for the Hamming metric by minimizing a margin based triplet loss.

[REF8] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Sparse\_representation\_learning/BIBREF87\_57a07372e2a620d6ae920f74877eee5f61753a96.pdf Title: SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval Chunk of text: All use the max score of passages in the document as the document score at the query time. Model MRR@10 R@1000 Query-tf 25.7 94.2 Query-neural-symmetric 26.4 94.7 Query-neural-asymmetric 25.4 94.2 Table 4: Performances of our model with different query representation strategies on our new Dev Set of MSMARCO passage retrieval. the representations of passages for document ranking. The first one represents the document as a sum of the passage representations while the second one uses a decayed weighted sum. The PassageRetrievalMax does not represent the document but just calculates the scores of passages in the document and choose the maximum score as the score of the document for ranking. Table 3 shows the ranking performance of baselines and our models. Here we only report the results of PassageRetrievalMax of our models.

[REF9] - paperID: ./papers\_pdf/paper\_section/Learned\_Sparse\_Retrieval-Sparse\_representation\_learning/BIBREF94\_0c57dcf959ead9530f9ec3ebe0dd58de42a3e8af.pdf Title: Expansion via Prediction of Importance with Contextualization Chunk of text: Errors are back-propagated through the entire BERT model with a learning rate of 2 × 10−5 with the Adam optimizer . We train in batches of 16 triples using gradient accumulation, and we evaluate the model on a validation set of 200 random queries from the development set every 512 triples. The optimal training iteration and re-ranking cutoff threshold is selected using this validation set. We roll back to the top-performing model after 20 consecutive iterations (training iteration 42) without improvement to Mean Reciprocal Rank at 10 (MRR@10). Baselines and Evaluation. We test our approach by re-ranking the results from several first-stage rankers. We report the performance using MRR@10, the official evaluation metric, on the MSMARCO passage ranking Dev set.

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