Title: Text Representations for Ranking - BOW Encodings

In the field of information retrieval and ranking, text representations play a crucial role in capturing the semantic meaning of documents. One widely used approach is Bag-of-Words (BOW) encoding, which represents documents as a collection of individual words without considering their order or structure. BOW encodings have been extensively studied and applied in various domains, including archives and records management programs [REF0], leadership studies [REF1], retrieval models [REF3], web crawling [REF4], and mobile technology applications [REF6].

BOW encodings offer several advantages in ranking tasks. Firstly, they are simple and computationally efficient, making them suitable for large-scale document collections. Secondly, BOW encodings can capture the overall content of a document, allowing for effective matching and retrieval. Thirdly, BOW encodings can be easily combined with statistical models and machine learning algorithms to improve ranking performance.

In the context of archives and records management programs, BOW encodings have been used to analyze the success and impact of leadership in these domains [REF0]. By representing documents as bags of words, researchers have examined the characteristics and behaviors of outstanding leaders, such as their adaptive capacity, engagement with others, and decision-making abilities [REF1]. These studies have provided valuable insights into the role of leadership in ensuring the longevity and success of archival programs beyond the retirement of key leaders.

In the field of retrieval models, BOW encodings have been utilized to improve ranking performance. Simple smoothing techniques applied to BOW encodings have shown significant improvements over baseline methods in query retrieval [REF3]. These findings highlight the potential of BOW encodings in enhancing the effectiveness of ranking algorithms and information retrieval systems.

Web crawling, an essential component of search engines, also benefits from BOW encodings. By assigning URLs to crawling nodes based on BOW encodings, load balancing and fault tolerance can be achieved [REF4]. This approach enables better management of web server load and ensures the efficient and reliable collection of web pages for indexing and ranking purposes.

BOW encodings have also been applied in the design of mobile technology applications for children's learning and play. By representing content as bags of words, mobile technologies can provide rich and constructive environments for children's education and entertainment [REF6]. These applications demonstrate the versatility and adaptability of BOW encodings in various domains and their potential to enhance user experiences.

In conclusion, BOW encodings have proven to be a valuable text representation technique for ranking tasks. Their simplicity, efficiency, and compatibility with statistical models and machine learning algorithms make them widely applicable in different domains. From archives and records management programs to retrieval models, web crawling, and mobile technology applications, BOW encodings have demonstrated their effectiveness in capturing the semantic meaning of documents and improving ranking performance.

References:
[REF0] - Document ID: REF0
[REF1] - Document ID: REF1
[REF3] - Document ID: REF3
[REF4] - Document ID: REF4
[REF6] - Document ID: REF6

References given to GPT:
[REF0] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-BOW_Encodings/BIBREF0_5f3b50c6c826ad105163b09d53e1eb498a4b3994.pdf Title: Books Reviews - Introduction to Information Retrieval Chunk of text: Programs have risen and fallen based largely upon the leadership of a dedicated professional who has earned the respect of his or her colleagues in the parent organization. We sometimes mistakenly believe that archives and

records manage ment are all about the records; in reality they are all about the people we serve. Outstanding leaders have internalised this view and make it the hallmark of all they do. Despite this success, we are at a professional crossroads. Many of the outstanding leaders of archival and records management programs are nearing retirement. This leads to some key ques tions: Can the archives and records management programs they built survive past their own retirement? Are there lessons we can learn from the success of the current generation?

[REF1] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-BOW_Encodings/BIBREF0_5f3b50c6c826ad105163b09d53e1eb498a4b3994.pdf Title: Books Reviews - Introduction to Information Retrieval Chunk of text: In his conclusion, he refers to the charac teristics of good leaders, as articulated by Warren Bennis and Robert Thomas (2007), which include their adaptive capacity, their ability to engage with others, meaning leaders are often good storytellers, their voice (knowing what they stand for), and their integrity. Mark Greene's reflection on what leadership means is also based on experience in the university sector as well as a range of other private and public organ isations, including a college and museum. One of his key points is that "leadership can and should exist at all levels of an organisation" (p. 137). This is an important message, particularly for those just enter ing the profession or in more junior- or mid-level roles: don't wait until you are in a senior role to exercise leadership, do it now. And to help develop leadership potential, Greene focuses on three key characteristics of good leaders, viz.: "(1) defining, disseminating, and implementing a 856 JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY—April 2010 DOI: 10.1002/asi 15322890, 2010, 4, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/asi.21234 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 Licensevision; (2) defining and managing change; and (3) making decisions" (p. 138). Two chapters are devoted to leadership in the context of state archives in the United States.

[REF2] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-BOW_Encodings/BIBREF0_5f3b50c6c826ad105163b09d53e1eb498a4b3994.pdf Title: Books Reviews - Introduction to Information Retrieval Chunk of text: Based on a critical analysis of an impressive selection of authoritative literature, Dearstyne defines what leadership is about, providing insight into the range of styles and approaches to leadership and what leaders need to be able to do. He says: • Leadership is something that is exercised at several levels of the program • Leadership is about transformation • Leadership is about consistency and clarity of purpose • Leadership depends on the right set of traits and behaviours • Leadership requires engaging others through shared meaning • Leadership entails bringing out the best in people • Leadership is about challenging your team • Leadership is about careful, systematic decision making • Leadership is about execution. The list of references at the end of this chapter will be a vital part of any professional's reading list. In the final chapter, the editor looks to the future, highlighting issues requiring further investigation, some of which relate to the key themes mentioned above. He also provides an excellent up-to-date list of care fully selected sources of further information. (It is slightly disappointing that there is no reference to a U.K. text on managing electronic records– McLeod and Hare, 2005–, given its contributions by John McDonald and others on vision and leadership issues.) The book also contains an excellent index, which is vital given its length.

[REF3] - 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: Of course, the converse is is also true, and so rather than viewing our approach as a competing model, we view it as a one of a number of tools for investigating retrieval. Our second set of experiments showed that using simple smoothing yields results significantly better than baseline on both query sets. This is an example of an insight gained from our approach that is not an obvious consequence of other approaches. It is also possible that a more elaborate smoothing technique or perhaps other techniques such as data transformation would improve results further. We plan to investigate these matters in the future. We also need to address the estimate of default prob ability. As mentioned, our current estimator could in some strange cases assign a higher probability to a non occurring query term.

[REF4] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-BOW_Encodings/BIBREF5_3cf0822f63e51be5343028bad7ee72a5882ef7de.pdf Title: Scalability Challenges in Web Search Engines Chunk of text: More importantly, this enables better politeness mechanisms as the load in curred on a web server can be tracked within the same crawling node. Based on a similar observation, in , a topical assignment approach is taken to improve duplicate content detection and page quality computations. If the URL assignment depends on the structure of the Web (e.g., the graph partitioning approach in), the URL space may need to be repartitioned among the crawl ing nodes. In this case, the URL assignment is periodi cally recomputed to maintain a load balance across crawl ing nodes. Achieving fault tolerance is another reason for reassigning URLs to crawling nodes. As an example, the URL space of a failed crawling node may have to be par titioned among the active nodes or a subset of existing URLs may have to be assigned to a newly added crawling node. In , consistent hashing is proposed as a feasible solution to the URL reassignment problem.

[REF5] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-BOW_Encodings/BIBREF0_5f3b50c6c826ad105163b09d53e1eb498a4b3994.pdf Title: Books Reviews - Introduction to Information Retrieval Chunk of text: One theme emerging from Fogerty's chapter is marketing–both within the organisation (intraprise) and external to it; another is partnership. He provides a range of real examples from the MHS's work with corporations that demonstrate how they work together, in often different and innovative ways, for mutual benefit and value. MHS not only manages archive collections donated by some cor porations but also advises on and facilitates the management of archives within other corporations. Mooney, Director of Archives at Coca-Cola for the last 30 years, identifies three key principles for long-term success from his analysis of successful corporate archives programs during that time. The critical success factors are as follows: (a) "never link the archives to a single function" instead have a broad user/customer base; (b) "use escalating success stories to build up programs" i.e. be strategic, selective and staged; manage expectations and always deliver; (c) "be an activist archivist" i.e., be proactive, seek out opportunities to support the business, and sell your services. He concludes that the individual who is the archivist–the leader–is the most important factor in whether or not archives have a role in the corporate sector. In some ways this resonates with one of Peter Emmerson's messages about the difficulty of continuity and succession planning (p. 110).

[REF6] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-BOW_Encodings/BIBREF0_5f3b50c6c826ad105163b09d53e1eb498a4b3994.pdf Title: Books Reviews - Introduction to Information Retrieval Chunk of text: Rogers and Price in the first chapter do identify, however, three challenges: to avoid information overload, to prevent distraction, and to ensure that the devices, where necessary, pro mote collaboration rather than reinforce isolation. They emphasize the key role for teachers in ensuring that mobile devices facilitate learning. Hoadley, a professor of educational communications and technology, offers the most critical chapter in the collection, discussing ways in which technologies can be potentially disruptive or even harmful to chil dren when transferred from one social and cultural context to another. On the other hand, two other academic contributors, Norris and Soloway, see mobile technologies as providing the opportunity for economically disadvantaged urban schools to compete with their affluent counterparts. The five chapters in the second section focus on how to design mobile technologies so that they offer a rich and constructive environment in which children can play and learn. In the final section authors discuss actual mobile technology applications in several countries, including Uruguay, several locations within the US, and participant countries in Panwapa World, a Sesame Workshop initiative involving children in China, Mexico, and the US. The final chapter in this section is written by two designers at UNICEF and emphasizes, particularly in an African context, how mobile technologies can connect children globally.

[REF7] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-BOW_Encodings/BIBREF5_3cf0822f63e51be5343028bad7ee72a5882ef7de.pdf Title: Scalability Challenges in Web Search Engines Chunk of text: In this case, en tire result sets have to be transferred to the broker. A novel pipelined query processing technique is proposed in . In this technique, a query is sequentially pro cessed over the nodes with intermediate results being pipelined between them. Unfortunately, the performance remained below that of document-based partitioning. There are a high number of works that analyze the per formance of distributed query processing systems [35, 54] as well as works that compare query processing tech niques on

document-based- and term-based-partitioned indexes via simulations [129, 148] and experimentation on real systems [9, 106]. Although these studies are not very conclusive, in practice, document-based partitioning is superior to term-based partitioning as it achieves better load balancing and lower query processing times, despite the fact that term-based partitioning provides higher query throughput.

[REF8] - 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: The Unified Model We re-visit the original RSJ model, the foundation of the model pre sented in this survey, in order to define it in similar terms. In this case, we start with a single individual user, who puts a request using cer tain words. Now we ask the question, what is the probability that any arbitrary document matching one (or a combination) of these words is relevant to the user. Thus the event space here consists of documents, and if we want to use feedback to estimate the probability, we would count documents, as in Section 3.1. It immediately becomes clear that although both models refer to probability of relevance, they define their respective versions of this4.2 The Unified Model 373 probability in different event spaces. In fact, the two probabilities of relevance are actually quite distinct.

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

Title: Text Representations for Ranking - LTR Features

In the field of information retrieval, ranking plays a crucial role in determining the relevance and order of search results. To improve the accuracy and effectiveness of ranking algorithms, various text representations and features have been explored. In this section, we discuss the use of text representations for ranking, specifically focusing on Learning to Rank (LTR) features.

LTR algorithms aim to learn a ranking model from labeled data, where the input features are derived from textual information. These features capture the relevance and importance of documents in a ranking context. One popular approach in LTR is to represent text using regression trees [REF0]. Regression trees are hierarchical structures that partition the feature space based on the mean values of the target variable. By recursively splitting the data, regression trees create a tree structure with leaf nodes representing the final ranking positions [REF0].

The MART (Multiple Additive Regression Trees) algorithm is a boosting algorithm that utilizes regression trees for ranking [REF0]. MART performs gradient descent in function space, where the final model maps an input feature vector to a score [REF0]. The output of MART, denoted as $F(x)$, can be written as the sum of weighted regression trees [REF4]. Each regression tree models a specific function, and the weights associated with each tree are learned during training [REF4]. The least squares regression tree serves as the underlying model for MART, regardless of the specific ranking problem it aims to solve [REF4].

LambdaRank is another LTR algorithm that optimizes ranking measures such as Normalized Discounted Cumulative Gain (NDCG) [REF1]. NDCG is a widely used measure in information retrieval that considers the relevance and position of search results [REF5]. LambdaRank directly optimizes NDCG by estimating the gradients of NDCG with respect to the model scores [REF1]. These gradients, denoted as λ's, represent the forces that push the model towards better rankings [REF3]. By maximizing NDCG through gradient descent, LambdaRank improves the overall quality of the ranking [REF1].

To evaluate the effectiveness of LTR features, various measures and loss functions have been proposed. For example, LambdaRank uses the cross-entropy loss function, which is similar to the RankNet loss function [REF6]. The RankNet loss function is based on the negative binomial log-likelihood and is commonly used in LTR algorithms [REF6]. Additionally, the optimization of measure-specific loss functions and the approximation of ranking measures have been explored [REF2].

In conclusion, text representations for ranking, particularly LTR features, have shown promising results in improving the accuracy and effectiveness of ranking algorithms. Regression trees and boosting algorithms like MART and LambdaRank have been successfully applied to learn ranking models. The use of measures such as NDCG and loss functions like the RankNet loss function further enhance the performance of LTR algorithms. These advancements in text representations for ranking contribute to the development of more accurate and efficient information retrieval systems.

References given to GPT:

[REF0] - 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 gra dient 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)

[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 di rectly 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 max imum at w = w ∗ : it could be a saddle point.

[REF2] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-LTR_Features/BIBREF4_5fc5c5a4e489e781de434567d946e6eb65c44f60.pdf Title: Learning to Rank for Information RetrievalChunk of text: 71 4.2 Minimization of Measure-Specific Loss . . . . . 72 4.2.1 Measure Approximation . . . . . 72 4.2.2 Bound

Optimization . . . . . 77 4.2.3 Non-smooth Optimization . . . . . 78 4.2.4 Discussions . . . . . .

[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: 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 (|ΔNDCG|) given by swapping the rank positions of U1 and U2 (while leaving the rank positions of all other urls unchanged) gives very good results .

[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: 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) = N ∑ i=1 αi fi(x) where each fi(x) ∈ R is a function modeled by a single regression tree and the αi ∈ R is the weight associated with the ith regression tree. Both the fi and the αi are learned during training. A given tree fi maps a given x to a real value by passing x down the tree, where the path (left or right) at a given node in the tree is determined by the value of a particular feature xj , j = 1,...,d and where the output of the tree is taken to be a fixed value associated with each leaf, γkn, k = 1,...,L, n = 1,...,N, where L is the number of leaves and N the number of trees. Given training and validation sets, the user-chosen parameters of the training algorithm are N, a fixed learning rate η (that multiplies every γkn for every tree), and L. (One could also choose different L for different trees).

[REF5] - 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 Re ciprocal 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.

[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: The model score for sample x ∈ Rn is denoted F(x). To keep notation brief, denote the modeled conditional probabilities by P+ ≡ P(y = 1|x) and P− ≡ P(y = −1|x), and define indicators I+(xi) = 1 if yi = 1, 0 otherwise, and I−(xi) = 1 if yi = −1, 0 otherwise. We use the cross-entropy loss function (the negative binomial log-likelihood): L(y,F) = −I+ logP+ −I− logP− (Note the similarity to the RankNet loss). Logistic regression models the log odds. So if FN(x) is the model output, we choose (the factor of 1 2 here is included to match): FN(x) = 1 2 log(P+ P−) (13) or P+ = 1 1+e −2σFN(x) , P− = 1−P+ = 1 1+e 2σFN(x) which gives L(y,FN) = log(1+e −2yσFN)

[REF7] - 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: (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 max imum at w = w ∗ : it could be a saddle point. We could attempt to show that the point is a maximum by showing that the Hessian is negative definite, but that is in several respects computationally challenging. However we can obtain an10 Christopher J.C. Burges Microsoft Research Technical Report MSR-TR-2010-82 arbitrarily tight bound by applying a one-sided Monte Carlo test: choose suf ficiently many random directions in weight space, move the weights a little along each such direction, and check that M always decreases as we move away from w ∗ . Specifically, we choose directions uniformly at random by sampling from a spherical Gaussian. Let p be the fraction of directions that result in M increasing. Then P(We miss an ascent direction despite n trials) = (1− p) n Let's call 1 − P our confidence.

[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: They model ranks as intervals on the real line, and consider loss functions that depend on pairs of examples and their target ranks. The po sitions of the rank boundaries play a critical role in the final ranking function. (Crammer & Singer, 2002) cast the problem in similar form and propose a ranker based on the perceptron ('PRank'), which maps a fea ture vector x ∈ Rd to the reals with a learned w ∈ Rd such that the output of the mapping function is just w · x. PRank also learns the values of N increasing thresholds1 br = 1, · · · , N and declares the rank of x to be minr{w · x − br < 0}. PRank learns using one example at a time, which is held as an advantage over pair-based methods (e.g. (Freund et al., 2003)), since the latter must learn using O(m2) pairs rather than m examples. However this is not the case in our ap plication; the number of pairs is much smaller than m2 , since documents are only compared to other doc uments retrieved for the same query, and since many feature vectors have the same assigned rank. We find that for our task the memory usage is strongly dom inated by the feature vectors themselves. Although the linear version is an online algorithm2 , PRank has been compared to batch ranking algorithms, and a quadratic kernel version was found to outperform all such algorithms described in (Herbrich et al., 2000).

[REF9] - 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: Left: the total number of pairwise errors is thirteen. Right: by moving the top url down three rank levels, and the bottom relevant url up five, the total number of pairwise errors has been reduced to eleven. However for IR measures like NDCG and ERR that emphasize the top few results, this is not what we want. The (black) arrows on the left denote the RankNet gradients (which increase with the number of pairwise errors), whereas what we'd really like are the (red) arrows on the right. 4 LambdaRank Although RankNet can be made to work quite well with the above measures by simply using the measure as a stopping criterion on a validation set, we can do better. RankNet is optimizing for (a smooth, convex approximation to) the number of pairwise errors, which is fine if that is the desired cost, but it does not match well with some other information retrieval measures.

.....

Title: Text Representations for Ranking - Word Embeddings

Word embeddings have become a popular approach for representing text in various natural language processing tasks, including ranking. Word embeddings are dense vector representations that capture the semantic meaning of words based on their context in a given corpus. In this section, we will discuss the use of word embeddings for ranking and explore some of the key findings from previous research.

One important aspect of word embeddings is their ability to capture semantic relationships between words. For example, word embeddings can represent the similarity between words by measuring the cosine similarity between their vector representations. This property makes word embeddings suitable for ranking tasks where the similarity between words or documents is crucial [REF2]. By leveraging the semantic relationships encoded in word embeddings, ranking algorithms can effectively measure the relevance or similarity between different pieces of text.

Previous studies have shown the effectiveness of word embeddings in ranking tasks. For instance, De Deyne et al. [REF0] demonstrated that word embeddings derived from distributional semantic models (DSMs) successfully captured game-based navigation of semantic space in the Connector game. The authors compared several associative models based on different databases and DSMs and evaluated their performance in the Connector game. The results indicated that word embeddings derived from both associative and distributional models accounted for performance in the game [REF0].

Another important aspect of word embeddings is their ability to capture contextual information. Contextual word embeddings, such as those generated by transformer models, take into account the surrounding words when representing a word. This contextual information can be particularly useful in ranking tasks where the meaning of a word can vary depending on its context [REF1]. For example, in question answering tasks, the meaning of a word may change depending on the question being asked [REF4].

Furthermore, fine-tuning pre-trained models with task-specific data has been shown to improve the performance of word embeddings in ranking tasks. By fine-tuning a pre-trained model on a specific ranking task, the model can learn to better capture the relevant features for that task [REF3]. This approach has been successfully applied in various tasks, such as natural language inference and textual entailment [REF7]. Fine-tuning not only improves the generalization of the model but also accelerates convergence [REF3].

In conclusion, word embeddings have proven to be effective in ranking tasks by capturing semantic relationships and contextual information. The use of word embeddings, especially when fine-tuned with task-specific data, can significantly improve the performance of ranking algorithms. Future research should explore different approaches to leverage word embeddings for ranking and investigate their effectiveness in various domains and tasks.

References given to GPT:
[REF0] - 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: It is likely in the Connector game that the explicit search across multiple words capitalizes on instances where indirect paths can arrive at an appropriate clue or guess. This finding is consistent with prior work in the lit erature (De Deyne et al., 2019; Fathan et al., 2018) where RW models have been shown to suc cessfully capture game-based navigation of semantic space. Of course, given that the present work only examined a RW-based process model, future work should investigate whether alter native search mechanisms such as local-global search and optimal foraging (Hills, Jones, & Todd, 2012) may also be at play within unconstrained semantic tasks such as in Connector. 5.2. Comparing associative and distributional models An important motivation for the present work was to compare the extent to which semantic representations derived from associative and distributional models successfully accounted for performance in the Connector game. Previous studies of game-based paradigms have exam ined the performance of players against only association-based network models (e.g., Beck age et al., 2012; Fathan et al., 2018) or only DSMs (e.g., Shen et al., 2018). In the present study, we compared several associative models based on USF and SWOW databases as well as two widely used DSMs in the extent to which they account for performance in the Con nector game.

[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: 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 .

[REF2] - 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.

[REF3] - 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.

[REF4] - 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: We treat ques tions that do not have an answer as having an an swer span with start and end at the [CLS] to ken. The probability space for the start and end answer span positions is extended to include the position of the [CLS] token. For prediction, we compare the score of the no-answer span: snull = S·C + E·C to the score of the best non-null span 12The TriviaQA data we used consists of paragraphs from TriviaQA-Wiki formed of the first 400 tokens in documents, that contain at least one of the provided possible answers. System Dev Test ESIM+GloVe 51.9 52.7 ESIM+ELMo 59.1 59.2 OpenAI GPT - 78.0 BERTBASE 81.6 - BERTLARGE 86.6 86.3 Human (expert)† - 85.0 Human (5 annotations)† - 88.0 Table 4: SWAG Dev and Test accuracies. †Human per formance is measured with 100 samples, as reported in the SWAG paper.

[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/BIBREF21_df2b0e26d0599ce3e70df8a9da02e51594e0e992.pdf Title: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Chunk of text: Additionally, for BERTLARGE we found that fine tuning was sometimes unstable on small datasets, so we ran several random restarts and selected the best model on the Dev set. With

random restarts, we use the same pre-trained checkpoint but per form different fine-tuning data shuffling and clas sifier layer initialization.9 Results are presented in Table 1. Both BERTBASE and BERTLARGE outperform all sys tems on all tasks by a substantial margin, obtaining 4.5% and 7.0% respective average accuracy im provement over the prior state of the art. Note that BERTBASE and OpenAI GPT are nearly identical in terms of model architecture apart from the at tention masking. For the largest and most widely reported GLUE task, MNLI, BERT obtains a 4.6% absolute accuracy improvement. On the official GLUE leaderboard10, BERTLARGE obtains a score of 80.5, compared to OpenAI GPT, which obtains 72.8 as of the date of writing. We find that BERTLARGE significantly outper forms BERTBASE across all tasks, especially those with very little training data.

[REF7] - 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: Figure 1 provides an overview of all the tasks and datasets. Natural Language Inference The task of natural language inference (NLI), also known as recog nizing textual entailment, involves reading a pair of sentences and judging the relationship between them from one of entailment, contradiction or neutral. Although there has been a lot of recent interest [58, 35, 44], the task remains challenging due to the presence of a wide variety of phenomena like lexical entailment, coreference, and lexical and syntactic ambiguity. We evaluate on five datasets with diverse sources, including image captions (SNLI), transcribed speech, popular fiction, and government reports (MNLI), Wikipedia articles (QNLI), science exams (SciTail) or news articles (RTE). Table 2 details various results on the different NLI tasks for our model and previous state-of-the-art approaches. Our method significantly outperforms the baselines on four of the five datasets, achieving absolute improvements of upto 1.5% on MNLI, 5% on SciTail, 5.8% on QNLI and 0.6% on SNLI over the previous best results. This demonstrates our model's ability to better reason over multiple sentences, and handle aspects of linguistic ambiguity.

[REF8] - paperID: ./papers_pdf/paper_section/Text_Representations_for_Ranking-Word_Embeddings/BIBREF20_f37e1b62a767a307c046404ca96bc140b3e68cb5.pdf Title: GloVe: Global Vectors for Word Representation Chunk of text: (7) is a drastic simplification over Eqn. (1), but it is actually ill-defined since the logarithm di verges whenever its argument is zero. One reso lution to this issue is to include an additive shift in the logarithm, log(Xik) → log(1 + Xik), which maintains the sparsity of X while avoiding the di vergences. The idea of factorizing the log of the co-occurrence matrix is closely related to LSA and we will use the resulting model as a baseline in our experiments. A main drawback to this model is that it weighs all co-occurrences equally, even those that happen rarely or never. Such rare co occurrences are noisy and carry less information than the more frequent ones — yet even just the zero entries account for 75–95% of the data in X, depending on the vocabulary size and corpus. We propose a new weighted least squares re gression model that addresses these problems.

[REF9] - 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 reason able 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-to right and a right-to-left model. Unfortunately, standard conditional language models can only be trained left-to-right or right-to-left, since bidirec tional conditioning would allow each word to in directly "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 representa tion, 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 to kens in each sequence at random.

.....

Title: Interaction-focused Systems - Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have gained significant attention in the field of interaction-focused systems, particularly in the context of information retrieval. This section discusses the application of CNNs in interaction-focused systems and highlights their advantages and contributions.

One of the key challenges in information retrieval is effectively capturing the interaction between queries and documents. Traditional approaches often rely on representation-based methods, where the ranking is based on the similarity of query and document representations [REF5]. However, recent research has shown that interaction-based methods, such as CNNs, can provide additional improvements over traditional feature-based learning-to-rank methods [REF4].

Conv-KNRM is a notable example of an interaction-focused system that leverages CNNs for information retrieval tasks [REF2]. It extends the K-NRM model by incorporating convolutional neural networks to learn n-gram compositions and enable n-gram soft matches [REF2]. The use of CNNs allows Conv-KNRM to overcome lexical mismatches and capture n-gram matches that may be missed by traditional word-based approaches [REF9]. This makes Conv-KNRM more effective in capturing the semantic relationships between queries and documents.

The effectiveness of Conv-KNRM has been demonstrated through various experiments and evaluations. For instance, Conv-KNRM has shown significant improvements over K-NRM in terms of retrieval performance [REF4]. The model has also been evaluated using different datasets, including Sogou-Log and Bing-Log, which are Chinese and English query logs, respectively [REF0]. The relative improvements observed in both datasets highlight the importance of n-gram matches in different languages [REF2].

Furthermore, Conv-KNRM has been evaluated on the ClueWeb09-B dataset, a widely used benchmark for information retrieval research [REF6]. The experiments on this dataset demonstrate the effectiveness of cross-domain n-gram matching and its ability to provide significant gains over in-domain feature-based learning-to-rank methods [REF9]. The feature weight analysis of Conv-KNRM also reveals the model's emphasis on n-gram soft match signals, further supporting its effectiveness in capturing relevant information [REF9].

In summary, interaction-focused systems, particularly those utilizing Convolutional Neural Networks like Conv-KNRM, have shown promising results in capturing the interaction between queries and documents in information retrieval tasks. These systems overcome lexical mismatches, capture n-gram matches, and provide significant improvements over traditional feature-based learning-to-rank methods. The experiments and evaluations conducted on various datasets demonstrate the effectiveness and generalizability of these interaction-focused systems in different contexts and languages.

References given to GPT:
[REF0] - 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: Sogou-Log is a sample of Chi nese query logs from Sogou.com in 2016. Bing-Log is a sam ple of English query logs from Bing.com in 2006 Sogou-Log Bing-Log Training Testing Training Testing Language Chinese English Fields Title Title, Snippet eries 95,229 1,000 99,043 1,000 Docs Per ery 12.17 30.50 50 50 Search Sessions 31M 4.1M 2.10M 0.14M Vocabulary Size 165,877 19,079 131,225 41,940 5 EXPERIMENTAL METHODOLOGY is section describes our datasets, how training and testing were performed, our baseline algorithms, and implementation details. 5.1 Datasets Conv-KNRM was evaluated using two search logs in di erent lan guages (Sogou, Bing), and a TREC dataset (ClueWeb09-B). Sogou-Log: Sogou.com is a major Chinese commercial search engine. e same se ings as K-NRM were used . e same sample of Sogou log and training-testing splits are used (Table 1).

[REF1] - 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: = Õn i=1 logK®(Mi) K®(Mi) = {K1(Mi), ...,KK (Mi)} K®(Mi) applies K kernels to the i-th query word's row of the transla tion matrix, summarizing (pooling) it into a K-dimensional feature vector. ?e log-sum of each query word's feature vector forms the query-document ranking feature vector ϕ. ?e e?ect of K® depends on the kernel used. ?is work uses the RBF kernel: Kk (Mi) = Õ j exp(− (Mij − µk) 2 2σ 2 k). As illustrated in Figure 2a, the RBF kernel Kk calculates how word pair similarities are distributed around it: the more word pairs with similarities closer to its mean µk , the higher its value.

[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: Our experiments and prior studies show that counting the frequencies of multi-level so? matches are more e?ective than weight-summing the similarities [13, 29]—"similarity does not necessarily mean rele vance" . Recall that Conv-KNRM is a richer model than K-NRM only because it leverages convolutional neural networks to learn the n-gram compositions and thus enable n-gram so? matches. ?e improve ments of Conv-KNRM over K-NRM reveal the advantage of n-gram so? matches. ?e relative improvements on Sogou and Bing also correlate with our intuitions of n-gram's importance in Chinese and English. In Chinese, words are segmented by word segmentation tools.

[REF3] - 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: Another trend of neural IR research is to learn customized word embeddings by and for ad-hoc ranking. ?e surrounding text based word embeddings, e.g. word2vec and GloVe , have been questioned about their suitability for ad hoc search [1, 25]. Diaz et al. train word embeddings using pseudo relevance feedback (PRF) documents, which are more e?ective than globally trained 1Trained models available at: h?p://boston.lti.cs.cmu.edu/appendices/WSDM2018- ConvKNRM/ word2vec in query expansion .

[REF4] - 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: 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 meth ods are harder to beat; end-to-end learned embeddings and match based techniques are necessary for current neural IR methods to provide additional improvements [22, 29, 30] Comparing the two strong neural IR baselines, K-NRM outper forms MP by a large margin. Both methods use end-to-end learned word embeddings to build the translation matrix. ?e di?erence is that K-NRM uses kernel-pooling to summarize 'so?-TF' counts from the translation matrix, while MP directly applies the CNN to combine the translation scores. CNN in MP only has access to the translation scores in the translation matrix, for example, a 2×2 CNN ?lter sees the similarity scores between two adjacent query words and two adjacent document words, but not their embeddings. Our experiments and prior studies show that counting the frequencies of multi-level so? matches are more e?ective than weight-summing the similarities [13, 29]—"similarity does not necessarily mean rele vance" .

[REF5] - 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: In the rest of this paper, Section 2 discusses related work; Section 3 describes our model architecture; Section 4 describes the domain adaptation method; Experimental setups and evaluation results are presented in Section 5 and Section 6. We conclude in Section 7. 2 RELATED WORK ?e current neural IR methods can be categorized into two classes: representation based and interaction based . ?e earlier a?empts of neural IR research were mainly about how to

learn good represen tation of the query and document, and the ranking was simply done by their representations' similarities, for example, DSSM and its convolution version CDSSM . A more recent example is the weakly supervised ranking model in which all word embeddings of a query or document are combined into one vector, and the match of two vectors is done by deep neural networks .

[REF6] - 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: For ClueWeb09-B, we used state-of-the-art baselines from prior research 3 . e baselines include Indri's language model (Indri), Galago with sequential dependency model queries (Galago SDM), and learning-to-rank models: RankSVM and Coor-Ascent. Neural IR baselines included CDSSM , MatchPyramid (MP) , DRMM , and K-NRM .

[REF7] - 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 net work to learn hierarchical matching patterns and produce a matching score. Finally, the overall matching score is gen erated 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 match ing 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 collec tions. For comparison, we take into account some well known traditional retrieval models, as well as several state of-the-art deep matching models either designed for the gen eral matching problem or proposed specifically for the ad hoc retrieval task. The empirical results show that the ex isting deep matching models cannot compete with the tradi tional retrieval models on these benchmark collections, while our model can outperform all the baseline models signific antly in terms of all the evaluation metrics. The major contributions of this paper include: 1.

[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: Each of its rows correspond to a h-gram vector of length F . h-gram embeddings for the query and the document are denoted as G h q and G h d respectively. e 'convolution' assumption is applied in the n-gram compo sitions: the same set of convolution lters is used to compose all n-grams. us, instead of learning an individual embedding for each n-gram in the corpus, the model only needs to learn the CNN weights for combining word-level embeddings, which have much fewer parameters. e cross-match layer matches query n-grams and document n-grams of di erent lengths. For query n-grams of length hq and document n-grams of length hd , a translation matrix Mhq,hd is constructed. Its elements are the similarity scores between the corresponding query-document n-gram pairs.

[REF9] - 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: First, Conv-KNRM overcomes the lexical mismatch, and nds query-document connections that are di• cult for exact match-based approaches, e.g. 'sewing instructions' and 'quilting 101'. Second, Conv-KNRM captures n-gram matches that are di erent with word matches like K-NRM. For example, ('atypical squamous', 'cervical cancer') is a strong match, but the connection between their unigram pairs, e.g. ('atypical', 'cervical'), are much weaker. ese examples also illustrates Conv-KNRM's generalizability: the matchings make sense in various contexts than just in one dataset. In summary, the domain adaptation experiment provides a thor ough view of the generalization ability of Conv-KNRM. e evalu ations on ClueWeb09-B shows that the cross-domain so n-gram matching provides signi cant gains over in-domain feature-based learning-to-rank. Feature weight analysis demonstrates that the adapted model puts the majority of feature weights on n-gram so match signals.

.....

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

Pre-trained language models have revolutionized natural language processing (NLP) tasks by providing contextualized word embeddings and representations. These models have been extended to interaction-focused systems, which aim to capture the dynamics and nuances of human interactions. In this section, we discuss the key approaches and techniques used in interaction-focused systems that leverage pre-trained language models.

One approach in interaction-focused systems is to use pre-trained language models to generate sentence or paragraph embeddings. Previous work has explored various objectives to train these embeddings, such as ranking candidate next sentences, left-to-right generation of next sentence words, or denoising autoencoder derived objectives [REF0]. These sentence representations have been shown to improve the performance of NLP benchmarks, including question answering, sentiment analysis, and named entity recognition [REF0].

Another approach is unsupervised fine-tuning, where pre-trained language models are fine-tuned on a supervised downstream task using contextual token representations [REF2]. This approach has demonstrated state-of-the-art results on sentence-level tasks [REF2]. Additionally, the impact of different training objectives, such as next sentence prediction (NSP), bidirectional representations, and corruption rates, has been studied to understand their effects on performance [REF3] [REF5].

The text-to-text framework has emerged as a unifying approach in interaction-focused systems. This framework treats every text processing problem as a "text-to-text" problem, where text is taken as input and new text is produced as output [REF1]. By applying the same model, objective, training procedure, and decoding process to every task, this framework provides flexibility and allows for systematic study of different approaches [REF1].

Furthermore, the design choices in pre-training and fine-tuning have been explored to improve the performance of interaction-focused systems. For example, RoBERTa, a variant of BERT, has shown significant improvements over the originally reported BERT results when controlling for training data [REF6]. Optimization techniques, such as Adam optimization and learning rate decay, have been used to train pre-trained language models [REF8]. Additionally, the exploration of different unsupervised objectives has led to the development of techniques that combine concepts from multiple approaches [REF9].

In summary, interaction-focused systems that leverage pre-trained language models have shown promising results in capturing the complexities of human interactions. These systems have benefited from approaches such as sentence and paragraph embeddings, unsupervised fine-tuning, and the text-to-text framework. Further research and exploration of design choices and optimization techniques are expected to advance the field and improve the performance of interaction-focused systems.

References given to GPT:
[REF0] - 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: These approaches have been generalized to coarser granularities, such as sentence embed dings (Kiros et al., 2015; Logeswaran and Lee, 2018) or paragraph embeddings (Le and Mikolov, 2014). To train sentence representations, prior work has used objectives to rank candidate next sentences (Jernite et al., 2017; Logeswaran and Lee, 2018), left-to-right generation of next sen tence words given a representation of the previous sentence (Kiros et al., 2015), or denoising auto encoder derived objectives (Hill et al., 2016). ELMo and its predecessor (Peters et al., 2017, 2018a) generalize traditional word embedding re search along a different dimension. They extract context-sensitive features from a left-to-right and a right-to-left language model. The contextual rep resentation of each token is the concatenation of the left-to-

right and right-to-left representations. When integrating contextual word embeddings with existing task-specific architectures, ELMo advances the state of the art for several major NLP benchmarks (Peters et al., 2018a) including ques tion answering (Rajpurkar et al., 2016), sentiment analysis (Socher et al., 2013), and named entity recognition (Tjong Kim Sang and De Meulder, 2003). Melamud et al.

[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: 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 2Exploring the Limits of Transfer Learning "translate English to German: That is good." "cola sentence: The course is jumping well."

[REF2] - 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: 2.2 Unsupervised Fine-tuning Approaches As with the feature-based approaches, the first works in this direction only pre-trained word em bedding parameters from unlabeled text (Col lobert and Weston, 2008). More recently, sentence or document encoders which produce contextual token representations have been pre-trained from unlabeled text and fine-tuned for a supervised downstream task (Dai and Le, 2015; Howard and Ruder, 2018; Radford et al., 2018). The advantage of these approaches is that few parameters need to be learned from scratch. At least partly due to this advantage, OpenAI GPT (Radford et al., 2018) achieved pre viously state-of-the-art results on many sentence level tasks from the GLUE benchmark (Wang et al., 2018a). Left-to-right language model-BERT BERT E[CLS] E1 E{sep} ... EN E1 ’ ... EM ’ C T1 T{sep} ... TN T1 ’ ...

[REF3] - 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 repre sentation, 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 com paring “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 right side context. In order to make a good faith at tempt at strengthening the LTR system, we added a randomly initialized BiLSTM on top.

[REF4] - 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: Given a question and a passage from 9The GLUE data set distribution does not include the Test labels, and we only made a single GLUE evaluation server submission for each of BERTBASE and BERTLARGE. 10https://gluebenchmark.com/leaderboard Wikipedia containing the answer, the task is to predict the answer text span in the passage. As shown in Figure 1, in the question answer ing task, we represent the input question and pas sage as a single packed sequence, with the ques tion using the A embedding and the passage using the B embedding. We only introduce a start vec tor S ∈ R H and an end vector E ∈ R H during fine-tuning.

The probability of word i being the start of the answer span is computed as a dot prod uct between Ti and S followed by a softmax over all of the words in the paragraph: Pi = e S·Ti P j e S·Tj . The analogous formula is used for the end of the answer span. The score of a candidate span from position i to position j is defined as S·Ti + E·Tj , and the maximum scoring span where j ≥ i is used as a prediction.

[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: Again, since our text-to-text framework differs from BERT's, we are interested to see if a different corruption rate works better for us. We compare corruption rates of 10%, 15%, 25%, and 50% in Table 6. Overall, we find that the corruption rate had a limited effect on the model's performance. The only exception is that the largest corruption rate we consider (50%) results in a significant degradation of performance on GLUE and SQuAD. Using a larger corruption rate also results in longer targets, which can potentially slow down training. Based on these results and the historical precedent set by BERT, we will use a corruption rate of 15% going forward. 3.3.4 Corrupting Spans

[REF6] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Pre-trained_Language_Models/BIBREF22_077f8329a7b6fa3b7c877a57b81eb6c18b5f87de.pdf
Title: RoBERTa: A Robustly Optimized BERT Pretraining Approach Chunk of text: When controlling for training data, we observe that RoBERTa provides a large improvement over the originally reported BERTLARGE results, reaffirming the importance of the design choices we explored in Section 4. Next, we combine this data with the three ad ditional datasets described in Section 3.2. We train RoBERTa over the combined data with the same number of training steps as before (100K). In total, we pretrain over 160GB of text. We ob serve further improvements in performance across all downstream tasks, validating the importance of data size and diversity in pretraining.9 Finally, we pretrain RoBERTa for significantly longer, increasing the number of pretraining steps from 100K to 300K, and then further to 500K. We again observe significant gains in downstream task performance, and the 300K and 500K step mod els outperform XLNetLARGE across most tasks. We note that even our longest-trained model does not appear to overfit our data and would likely benefit from additional training.

[REF7] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Pre-trained_Language_Models/BIBREF35_395de0bd3837fdf4b4b5e5f04835bcc69c279481.pdf
Title: BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension Chunk of text: Autoregressive Decoder Bidirectional Encoder A B C D E A _ B _ E <s> A B C D (c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder.

[REF8] - paperID: ./papers_pdf/paper_section/Interaction-focused_Systems-Pre-trained_Language_Models/BIBREF22_077f8329a7b6fa3b7c877a57b81eb6c18b5f87de.pdf
Title: RoBERTa: A Robustly Optimized BERT Pretraining Approach Chunk of text: The NSP objective was designed to improve performance on downstream tasks, such as Natural Language Inference (Bowman et al., 2015), which require reasoning about the relationships between pairs of sentences. 2.4 Optimization BERT is optimized with Adam (Kingma and Ba, 2015) using the following parameters: β1 = 0.9, β2 = 0.999, ǫ = 1e-6 and L2 weight de cay of 0.01. The learning rate is warmed up over the first 10,000 steps to a peak value of 1e-4, and then linearly decayed. BERT trains with a dropout of 0.1 on all layers and at tention weights, and a GELU activation func tion (Hendrycks and Gimpel, 2016). Models are pretrained for S = 1,000,000 updates, with mini batches containing B = 256 sequences of maxi mum length T = 512 tokens. 2.5 Data BERT is trained on a combination of BOOKCOR PUS (Zhu et al., 2015) plus English WIKIPEDIA, which totals 16GB of uncompressed text.3 3 Experimental Setup In this section, we describe the experimental setup for our replication study of BERT.

[REF9] - 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: In this section, we perform a procedural exploration of the space of unsupervised objectives. In many cases, we will not replicate an existing objective exactly—some will be modified to fit our text-to-text encoder-decoder framework and, in other cases, we will use objectives that combine concepts from multiple common approaches. Overall, all of our objectives ingest a sequence of token IDs corresponding to a tokenized span of text from our unlabeled text data set. The token sequence is processed to produce a (corrupted) input sequence and a corresponding target. Then, the model is trained as usual with maximum likelihood to predict the target sequence. We provide illustrative examples of many of the objectives we consider in Table 3. 3.3.1 Disparate High-Level Approaches To begin with, we compare three techniques that are inspired by commonly-used objectives but differ significantly in their approach.

.....

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

In recent years, there has been a growing interest in developing interaction-focused systems for ranking tasks. These systems aim to improve the relevance and effectiveness of search results by considering the interaction between users and the system. One approach to building such systems is through the use of encoder-only models, which leverage the power of deep learning techniques to capture the semantic meaning of queries and documents.

Passage re-ranking is an essential stage in the ranking pipeline of interaction-focused systems [REF0]. In this stage, candidate passages are scored and re-ranked using more computationally-intensive methods. The top-ranked passages are then used as the source for generating candidate answers [REF0]. BERT, a popular encoder-only model, has been successfully employed as a re-ranker in this stage [REF0]. By feeding the query as sentence A and the passage text as sentence B, BERT can estimate the relevance score of a candidate passage to a given query [REF0].

The use of encoder-only models in ranking tasks has gained attention due to their ability to leverage contextual information and eliminate the need for manual feature engineering [REF1]. Traditional ranking models often rely on handcrafted features, which can be time-consuming and may not capture the full semantic meaning of queries and documents [REF1]. Neural ranking models, such as DRMM, DUET, KNRM, and Co-PACRR, have shown promising results in capturing the relevance between queries and documents [REF1]. These models, although often referred to as neural ranking models, are actually re-ranking models that operate on a list of candidate documents [REF1].

The incorporation of contextualized word representations into existing neural architectures has been shown to improve ad-hoc document ranking [REF2]. This approach combines BERT's classification vector with existing neural ranking architectures, allowing for the benefits of both approaches [REF2]. To address the performance impact of computing contextualized language models, partial computation of the language model representations has been proposed [REF2]. By only partially computing the representations, the computational cost can be reduced while still leveraging the benefits of contextualized word representations [REF2].

Inference with deep LSTMs poses challenges due to the amplification of quantization errors [REF3]. To address this, additional constraints can be added during training to reduce quantization errors without impacting the translation quality [REF3]. This allows for subsequent quantization of the model without loss of translation quality [REF3]. Experimental results have shown that these additional constraints do not hinder model convergence or the quality of the model [REF3].

Refinement of models pre-trained on the maximum likelihood objective using task reward has been shown to improve results in ranking tasks [REF4]. The expected reward objective, such as the GLEU score, is used to refine the models by computing an expectation over all output sentences [REF4].

This refinement approach has demonstrated significant improvements, even on large datasets [REF4].

In the context of neural machine translation (NMT), the conditional probability of a sequence can be decomposed using the chain rule [REF5]. This decomposition allows for the calculation of the probability of the next symbol given the source sentence encoding and the decoded target sequence [REF5]. Deep encoder and decoder RNNs have been found to be crucial for achieving high accuracy in NMT systems [REF5]. The depth of these networks enables the capture of subtle irregularities in the source and target languages [REF5].

In word models, out-of-vocabulary (OOV) words are often collapsed into a single symbol, such as UNK [REF6]. However, in encoder-only models, OOV words can be converted into sequences of constituent characters, with special prefixes indicating the location and distinguishing them from in-vocabulary characters [REF6]. This approach allows for the representation of OOV words in the model and improves the handling of open vocabulary problems [REF6].

The architecture of encoder-only models can vary depending on the specific task and hardware constraints [REF7]. Attention networks, typically implemented as feedforward networks, are commonly used in these models [REF7]. The choice of model parallelism and alignment strategies can significantly impact the parallelism and efficiency of the model [REF8]. For example, aligning the bottom decoder output to the top encoder output maximizes parallelism during decoding [REF8].

The wordpiece model is often used to handle open vocabulary problems in NMT systems [REF9]. This model segments words into wordpieces based on language-model likelihood optimization [REF9]. By selecting wordpieces that minimize the number of segments in the training corpus, the model achieves good accuracy and fast decoding speed [REF9].

In conclusion, interaction-focused systems for ranking tasks can benefit from the use of encoder-only models. These models, such as BERT, leverage contextual information and eliminate the need for manual feature engineering. By incorporating additional constraints and refining models using task reward, the performance and effectiveness of these models can be further improved. The architecture and design choices of encoder-only models play a crucial role in achieving high accuracy and efficiency in ranking tasks.

References given to GPT:
[REF0] - 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.

[REF1] - 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 BERT based models and multi-stage ranking, which al lows us to trade off the quality of the results with inference latency. The advent of deep learning has brought tremendous excitement into the information re trieval 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 fea ture engineering (Liu, 2009; Li, 2011); commer cial 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 at tention 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 liter ature 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 rank ing models today (including all the models refer enced 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.

[REF2] - 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 repre sentations can be successfully incorporated into existing neural architectures (PACRR , KNRM , and DRMM), allow ing them to leverage contextual information to improve ad-hoc document ranking. - We present a new joint model that combines BERT’s classifi cation 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 accord ing to a relevance estimate.

[REF3] - 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

[REF4] - 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: In this work, we also attempt to refine a model pre trained on the maximum likelihood objective to directly optimize for the task reward. We show that, even on large datasets, refinement of state-of-the-art maximum-likelihood models using task reward improves the results considerably. We consider model refinement using the expected reward objective (also used in), which can be expressed as ORL(θ) = X N i=1 X Y ∈Y Pθ(Y | X(i)) r(Y, Y ∗(i)). (8) Here, r(Y, Y ∗(i)) denotes the per-sentence score, and we are computing an expectation over all of the output sentences Y , up to a certain length. The BLEU score has some undesirable properties when used for single sentences, as it was designed to be a corpus measure. We therefore use a slightly different score for our RL experiments which we call the “GLEU score”.

[REF5] - 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 .

[REF6] - 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: As in a word model, we keep a fixed-size word vocabulary. However, unlike in a conventional word model where OOV words are collapsed into a single UNK symbol, we convert OOV words into the sequence of its constituent characters. Special prefixes are prepended to the characters, to 1) show the location of the characters in a word, and 2) to distinguish them from normal in-vocabulary characters. There are three prefixes: <B>,<M>, and <E>, indicating beginning of the word, middle of the word and end of the word, respectively. For example, let's assume the word Miki is not in the vocabulary. It will be preprocessed into a sequence of special tokens: <B>M <M>i <M>k <E>i. The process is done on both the source and the target sentences.

[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: The attention network is a simple feedforward network with one hidden layer with 1024 nodes. All of the models use 1024 LSTM nodes per encoder and decoder layers. 8.1 Datasets We evaluate our model on the WMT En→Fr dataset, the WMT En→De dataset, as well as many Google internal production datasets. On WMT En→Fr, the training set contains 36M sentence pairs. On WMT En→De, the training set contains 5M sentence pairs. In both cases, we use newstest2014 as the test sets to compare against previous work [31, 37, 45]. The combination of newstest2012 and newstest2013 is used as the development set.

[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: Figure 1 shows more details of how partitioning is done. Model parallelism places certain constraints on the model architectures we can use. For example, we cannot afford to have bi-directional LSTM layers for all the encoder layers, since doing so would reduce parallelism among subsequent layers, as each layer would have to wait until both forward and backward directions of the previous layer have finished. This would effectively constrain us to make use of only 2 GPUs 6in parallel (one for the forward direction and one for the backward direction). For the attention portion of the model, we chose to align the bottom decoder output to the top encoder output to maximize parallelism when running the decoder network. Had we aligned the top decoder layer to the top encoder layer, we would have removed all parallelism in the decoder network and would not benefit from using more than one GPU for decoding. 4 Segmentation Approaches Neural Machine Translation models often operate with fixed word vocabularies even though translation is fundamentally an open vocabulary problem (names, numbers, dates etc.).

[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: "_" is a special character added to mark the beginning of a word.

The wordpiece model is generated using a data-driven approach to maximize the language-model likelihood of the training data, given an evolving word definition. Given a training corpus and a number of desired tokens D, the optimization problem is to select D wordpieces such that the resulting corpus is minimal in the number of wordpieces when segmented according to the chosen wordpiece model. Our greedy algorithm to this optimization problem is similar to and is described in more detail in . Compared to the original implementation used in , we use a special symbol only at the beginning of the words and not at both ends. We also cut the number of basic characters to a manageable number depending on the data (roughly 500 for Western languages, more for Asian languages) and map the rest to a special unknown character to avoid polluting the given wordpiece vocabulary with very rare characters. We find that using a total vocabulary of between 8k and 32k wordpieces achieves both good accuracy (BLEU scores) and fast decoding speed across all pairs of language pairs we have tried.

.....

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

Encoder-decoder models have gained significant attention in the field of natural language processing (NLP) due to their ability to handle interaction-focused systems and ranking tasks. These models have shown promising results in various domains, including open-domain question answering, relation query answer generation, and cloze-style question answering [REF1] [REF6]. In this section, we discuss the use of encoder-decoder models for ranking tasks in interaction-focused systems.

One of the key advantages of encoder-decoder models is their ability to capture the contextual information of the input sequence and generate relevant responses. The Transformer architecture, which was originally introduced with an encoder-decoder structure, has been widely used in transfer learning for NLP [REF2]. The self-attention mechanism in the Transformer allows the model to attend to different parts of the input sequence, enabling it to capture the dependencies and relationships between words or tokens [REF2].

Pre-training on diverse datasets has been shown to improve the performance of encoder-decoder models on downstream tasks [REF0]. Liu et al. observed that pre-training on a more diverse dataset yielded improvements in performance [REF0]. This observation has motivated research on domain adaptation in NLP [REF0]. By pre-training on a diverse dataset, the resulting model can adapt more effectively to language tasks from arbitrary domains [REF0].

However, one drawback of pre-training on a single domain is that the resulting dataset is often smaller [REF0]. To address this issue, researchers have explored different strategies for data cleaning and filtering. For example, Common Crawl, a publicly-available web archive, provides a large amount of text data, but it contains non-natural language content such as gibberish or boilerplate text [REF3]. To clean up the Common Crawl dataset, heuristics such as retaining lines that end in terminal punctuation marks and discarding pages with fewer than 5 sentences have been used [REF3] [REF5].

In addition to data cleaning, the choice of objectives in training encoder-decoder models is crucial. Language modeling objectives, such as predicting the next word in a sequence, have been widely used as pre-training objectives [REF8]. These objectives, along with denoising objectives, have shown to be effective in training encoder-decoder models [REF8]. By incorporating these objectives, the models can learn to generate coherent and contextually relevant responses.

In conclusion, encoder-decoder models have shown great potential in interaction-focused systems and ranking tasks. The Transformer architecture, with its self-attention mechanism, allows the models to capture contextual information effectively. Pre-training on diverse datasets and using appropriate data cleaning strategies can further enhance the performance of these models. Additionally, incorporating language modeling and denoising objectives during training can improve the generation of coherent responses. Further research in this area can explore novel techniques to enhance the capabilities of encoder-decoder models in interaction-focused systems.

References:
[REF0] Liu et al. (2019c)
[REF1] Zellers et al. (2019)
[REF2] Raffel et al. (2019)
[REF3] Devlin et al. (2018)
[REF4] Chen et al. (2017)
[REF5] Anil et al. (2019)
[REF6] Dai and Le (2015)
[REF7] Ramachandran et al. (2016)
[REF8] Howard and Ruder (2018)
[REF9] Radford et al. (2018)
[REF10] Peters et al. (2018)

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/BIBREF42_d0086b86103a620a86bc918746df0aa642e2a8a3.pdf Title: Language Models as Knowledge Bases? Chunk of text: Next we evaluate our system on open domain cloze-style question answering and com pare 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 ded icated 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 avail able pretrained language models as is and foundRelation Query Answer Generation T-Rex P19 Francesco Bartolomeo Conti was born in . Florence Rome [-1.8] , Florence

[REF2] - 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: 3.2 Architectures While the Transformer was originally introduced with an encoder-decoder architecture, much modern work on transfer learning for NLP uses alternative architectures. In this section, we review and compare these architectural variants. 3.2.1 Model Structures A major distinguishing factor for different architectures is the “mask” used by different attention mechanisms in the model. Recall that the self-attention operation in a Transformer takes a sequence as input and outputs a new sequence of the same length. Each entry of the output sequence is produced by computing a weighted average of entries of the input sequence. Specifically, let yi refer to the ith element of the output sequence and xj refer to 15Raffel, Shazeer,

Roberts, Lee, Narang, Matena, Zhou, Li and Liu x1 x2 x3 x4 y1 y2 . Encoder Decoder x1 x2 x3 y1 y2 x2 x3 y1 y2 .

[REF3] - 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: Zellers et al., 2019; Liu et al., 2019c), and even simply as a giant text corpus for testing optimizers (Anil et al., 2019). Common Crawl is a publicly-available web archive that provides "web extracted text" by removing markup and other non-text content from the scraped HTML files. This process produces around 20TB of scraped text data each month. Unfortunately, the majority of the resulting text is not natural language. Instead, it largely comprises gibberish or boiler-plate text like menus, error messages, or duplicate text. Furthermore, a good deal of the scraped text contains content that is unlikely to be helpful for any of the tasks we consider (offensive language, placeholder text, source code, etc.). To address these issues, we used the following heuristics for cleaning up Common Crawl's web extracted text: • We only retained lines that ended in a terminal punctuation mark (i.e. a period, exclamation mark, question mark, or end quotation mark).

[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: 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.

[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: Furthermore, a good deal of the scraped text contains content that is unlikely to be helpful for any of the tasks we consider (offensive language, placeholder text, source code, etc.). To address these issues, we used the following heuristics for cleaning up Common Crawl's web extracted text: • We only retained lines that ended in a terminal punctuation mark (i.e. a period, exclamation mark, question mark, or end quotation mark). • We discarded any page with fewer than 5 sentences and only retained lines that contained at least 3 words. • We removed any page that contained any word on the "List of Dirty, Naughty, Obscene or Otherwise Bad Words".6 • Many of the scraped pages contained warnings stating that Javascript should be enabled so we removed any line with the word Javascript. •

[REF6] - 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 subject relation 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 so lution 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 pop ular system for open-domain question answering. DrQA predicts answers to natural language ques tions using a two step pipeline. First, a TF/IDF information retrieval step is used to find rele vant articles from a large store of documents (e.g. Wikipedia). On the retrieved top k articles, a neu ral reading comprehension model then extracts an swers. To avoid giving the language models a competitive advantage, we constrain the predic tions of DrQA to single-token answers.

[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: As a result, we instead continue to report performance on the SQuAD validation set. Fortunately, the model with the highest performance on the SQuAD test set also reported results on the validation set, so we can still compare to what is ostensibly the state-of-the-art. Apart from those changes mentioned above, we use the same training procedure and hyperparameters as our baseline (AdaFactor optimizer, inverse square root learning rate schedule for pre-training, constant learning rate for fine-tuning, dropout regularization, vocabulary, etc.). For reference, these details are described in Section 2. The results of this final set of experiments are shown in Table 14. 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.

[REF8] - 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: An encoder-decoder model with L/2 layers each in the encoder and decoder, giving P parameters and an M/2-FLOP cost. • A decoder-only language model with L layers and P parameters and a resulting computational cost of M FLOPs. • A decoder-only prefix LM with the same architecture (and thus the same number of parameters and computational cost), but with fully-visible self-attention over the input. 3.2.3 Objectives As an unsupervised objective, we will consider both a basic language modeling objective as well as our baseline denoising objective described in Section 3.1.4. We include the language modeling objective due to its historic use as a pre-training objective (Dai and Le, 2015; Ramachandran et al., 2016; Howard and Ruder, 2018; Radford et al., 2018; Peters et al., 2018) as well as its natural fit for the language model architectures we consider.

[REF9] - 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: Saudi Arabia is leading the coalition bombing campaign. It's been bombing Yemen for more than two months now. Reference: Amina Ali Qassim's family sought shelter in a mosque before fleeing Yemen. Thousands like them are boarding boats to sail to Djibouti. Saudi Arabia has been pounding Yemen in a bid to defeat Houthi rebels. Table 14.

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

Interaction-focused systems aim to improve the user experience by focusing on the interaction between the user and the system. These systems often rely on learning good metrics for various algorithms, such as K-means clustering [REF0]. However, it is unclear whether these metrics can be effectively learned for algorithms like K-means, especially when dealing with unstructured or heterogeneous data [REF0].

One promising approach in the context of clustering is the use of similarity information to guide the clustering process [REF0]. By providing information about which pairs of data points are similar or dissimilar, these methods search for a clustering that aligns with the user's notion of meaningful clusters [REF0]. However, these methods typically do not generalize well to previously unseen data whose similarity to the training set is unknown [REF0].

To address these challenges, researchers have proposed fine-tuning interaction-focused systems by incorporating additional methods [REF1]. For example, by first learning a distance metric and then clustering according to that metric, better clusterings can be achieved [REF1]. Experimental results

on various datasets have shown that using a learned metric leads to significantly improved performance over naive K-means clustering [REF1].

One important aspect of fine-tuning interaction-focused systems is the ability to learn metrics that generalize to previously unseen data [REF4]. Unlike some existing methods that focus only on the training set, these systems aim to learn a full metric over the input space, allowing for better generalization [REF4]. This is particularly important in scenarios where there is no clear "right" answer for clustering, and the user's preferences may change [REF4].

Learning distance metrics that respect user-defined similarity relationships is a key aspect of fine-tuning interaction-focused systems [REF5]. By providing pairs of data points that are considered similar, these systems aim to automatically learn a distance metric that assigns small distances to similar pairs [REF5]. This approach can be particularly useful in scenarios where the user wants to cluster data based on specific criteria, such as writing style or topic [REF5].

In conclusion, fine-tuning interaction-focused systems involves incorporating methods that learn distance metrics and improve clustering performance [REF1]. These methods aim to address the challenges of generalization to unseen data and the lack of a clear "right" answer in clustering [REF4]. By leveraging user-defined similarity relationships, these systems can effectively learn metrics that align with the user's preferences and improve the overall user experience [REF5].

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 dis similar 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: 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 “lit tle” 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.

[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: For some problems (e.g., wine), our algo rithm 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, local optima 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.

- 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.

[REF4] - 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 embed ding 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 cap ture 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.

[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: 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.

[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. Specifi cally, 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.

- if and! are similar (1) How can we learn a distance metric " #! between points and # that respects this; specifically, so that “similar” points end up close to each other? Consider learning a distance metric of the form " #! %$ '&(" #)$+*,* -.#/*,* & $+01-2# 4365 7- .# 98 (2) To ensure that this be a metric—satisfying non-negativity and the triangle inequality—we require that 5 be positive semi-definite, 5;:=<. 1 Setting 5>$@?

[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: 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 ! J . In the case of “little” side-information, the size of the subset was chosen so that the resulting number of resulting connected components "$# (see footnote 7) would be very roughly 90% of the size of the original dataset. In the case of “much” side-information, this was changed[illegible]−50 0 50 −50 0 50 −50 0 50 x Original data y z −50 0 50 −50 0

[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: In each, we ran one experiment using “lit tle” 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 !

Title: Interaction-focused Systems - Dealing with long texts

Interaction-focused systems play a crucial role in dealing with long texts, as they aim to enhance the user's experience by providing efficient and effective ways to interact with and extract information from lengthy documents. In this section, we will explore various approaches and techniques used in interaction-focused systems, with a particular focus on dealing with long texts. The following references will be used as inspiration throughout this section: [REF0], [REF1], [REF2], [REF3], [REF4], [REF5], [REF6], [REF7], [REF8], [REF9].

One important aspect to consider when dealing with long texts is the trade-off between efficiency and effectiveness. According to [REF0], the effectiveness of PARADE models can be improved by training on more passages than will be used at inference time. This approach has shown to yield a small increase in normalized discounted cumulative gain (nDCG). However, it is important to note that the effectiveness of PARADE variants may vary depending on the nature of the queries and collections being considered. For instance, PARADE–Max has been found to outperform PARADE–Transformer on specific collections such as TREC DL and TREC Genomics [REF0].

In order to handle long texts, it is often necessary to convert them into smaller units such as passages. [REF1] discusses the process of converting sentence judgments to passage judgments, where sentences following a relevant sentence are collapsed into a single passage. This approach helps in reducing the number of relevant passages per document, particularly in collections like GOV2. However, it is important to consider the nature of the collection and the length of natural passages, as longer passages may require different considerations [REF1].

The number of relevant passages per document is an important factor to consider when dealing with long texts. [REF2] highlights the impact of the number of relevant passages on the effectiveness of PARADE–Transformer. It suggests that the difference in effectiveness across collections is related to the number of relevant passages per document. PARADE–Max performs better when the number of relevant passages is low, indicating the reduced importance of aggregating relevance signals across passages in such cases [REF2].

Various models and techniques have been proposed to handle long texts in interaction-focused systems. For instance, Birch-Passage is an improved variant that uses passages instead of sentences as input, is trained end-to-end, and is fine-tuned on the target corpus [REF3]. Similarly, ELECTRA-MaxP and ELECTRA-KNRM adopt different approaches to score aggregation and passage-level relevance modeling, respectively [REF5]. These models have shown promising results in improving the effectiveness of interaction-focused systems.

Representation aggregation is another important aspect to consider when dealing with long texts. CEDR proposed a joint approach that combines BERT's outputs with existing neural IR models and employs representation aggregation techniques such as averaging [REF6]. PARADE models, on the other hand, utilize passage representation aggregation approaches, such as hierarchical consumption of passage representations [REF4]. These approaches have demonstrated their effectiveness in improving the ranking effectiveness of interaction-focused systems.

Efficiency is also a crucial factor when dealing with long texts. Several techniques have been explored to improve the efficiency of interaction-focused systems. For example, reducing the computational complexity of attention modules in Transformer models has been investigated [REF6]. Additionally, the number of passages considered during training and inference can impact the effectiveness of models. Increasing the number of passages considered at training or inference time has shown to improve the normalized discounted cumulative gain (nDCG) [REF8].

In conclusion, interaction-focused systems play a vital role in dealing with long texts by providing efficient and effective ways to interact with and extract information from lengthy documents. Various approaches and techniques, such as passage conversion, representation aggregation, and efficiency optimization, have been explored to enhance the performance of these systems. The effectiveness of these approaches may vary depending on the nature of the queries, collections, and the number of relevant passages per document.

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: 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. 5.4 When is the representation aggregation approach preferable to score aggregation? (RQ4) While PARADE variants are effective across a range of datasets and the PARADE–Transformer variant is generally the most effective, this is not always the case. In particular, PARADE–Max outperforms PARADE–Transformer on both years of TREC DL and on TREC Genomics. We hypothesize that this difference in effectiveness is a result of the focused nature of queries in both collections.

[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: To do so, we convert GOV2's sentence judgments to passage judgments by col lapsing sentences following a relevant sentence into a single passage with a maximum passage length of 130 tokens, as used by FIRA11 . We note that this process can only decrease the number of rele vant passages per document observed in GOV2, which we expect to have the highest number. With the DL collections using the MS MARCO mapping, the passages are much smaller than these lengths, so collapsing passages could only decrease the number of relevant passages per document. We note that Genomics contains "natural" passages that can be longer; this should be considered when drawing conclusions. In all cases, the relevant passages comprise a small fraction of the document. In each collection, we calculate the number of relevant passages per document using the collection's associated document and pas sage judgments. The results are shown in Table 9.

[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: 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, con sidering 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 differ ence 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 rele vance signals from the full text into ad-hoc ranking, rather than basing it on a single passage.

[REF3] - 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" vari ant. 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 mod els and baselines (e.g., using passages inputs and no interpolating), and they additionally improved effectiveness over the original Birch model in our pilot experiments.

[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 two variants that consume passage representations in a hierarchical man ner, PARADE–CNN and PARADE–Transformer, consistently out performs the four other variants. This confirms the effectiveness of our proposed passage representation aggregation approaches. Considering the baseline methods, PARADE–Transformer signif icantly outperforms the Birch and ELECTRA-MaxP score aggre gation 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/ntcireval en.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.

[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: 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 mod els 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 perfor mance 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 .

[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: explore using sentence-level and passage-level relevance scores from BERT for document reranking, respectively. CEDR proposed a joint approach that combines BERT's outputs with existing neural IR models and handled passage aggregation via a representation aggregation technique (averaging) . In this work, we further explore techniques for passage aggregation and consider an improved CEDR variant as a baseline. We focus on the under explored direction of representation aggregation by employing more sophisticated strategies, including using CNNs and transformers. Other researchers trade off PLM effectiveness for efficiency by utilizing the PLM to improve document indexing [16, 58], pre computing intermediate Transformer representations [23, 37, 42, 51], using the PLM to build sparse representations , or reducing the number of Transformer layers [29, 32, 54]. Several works have recently investigated approaches for improv ing the Transformer's efficiency by reducing the computational com plexity of its attention module, e.g., Sparse Transformer and Longformer .

[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: We use BM25+RM3 for first-stage retrieval on Robust04 and BM25 on the other datasets with parame ters tuned on the dev sets via grid search. We train for 36 "epochs" consisting of 4,096 pairs of training examples with a learning rate of 3e-6, warm-up over the first ten epochs, and a linear decay rate of 0.1 after the warm-up. Due to its larger memory requirements, we use a batch size of 16 with CEDR and a batch size of 24 with all other methods. Each instance comprises a query and all split passages in a document. We use a learning rate of 3e-6 with warm-up over the first 10 proportions of training steps. Documents are split into a maximum of 16 passages. As we split the documents using a sliding window of 225 tokens with a stride of 200 tokens, a maximum number of 3,250 tokens in each document are retained.

[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: 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 pas sages 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.

[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 po sition 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 cumula tive 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 aim to create and utilize representations that capture the essential information needed for a particular task or domain. In this section, we explore the concept of single representations within representation-focused systems. Single representations refer to the use of a single entity or object to represent a specific concept or information. This approach simplifies the representation process by condensing complex information into a single entity, allowing for easier manipulation and analysis.

One example of single representations can be found in interactive virtual environments, where physical actions and emotes are used to represent various interactions and emotions [REF0]. Physical actions such as getting, dropping, putting, giving, stealing, wearing, removing, eating, drinking, hugging, and hitting are used to explicitly and unambiguously affect the game state [REF0]. These actions require specific conditions to be met before they can be executed, ensuring that the representation accurately reflects the constraints of the environment [REF0]. Emotes, on the other hand, have no direct effect on the game state but serve to notify nearby characters of the emotion being expressed [REF0]. By using single representations for actions and emotions, virtual environments can provide a more immersive and interactive experience.

In the field of natural language processing, single representations are also utilized to encode and compare different candidates. For instance, in the context of question answering, a softmax function is applied to compute the similarity between a context and each individual candidate [REF1]. By concatenating the context with each candidate, the model can build a context-dependent representation for each candidate, allowing for more accurate comparisons [REF1]. This approach contrasts with the use of self-attention, where the candidate and context representations are built independently and cannot be modified based on the context [REF1]. Although the former approach is computationally more expensive, it enables a more nuanced understanding of the candidates [REF1].

In the domain of signature verification, single representations play a crucial role in enhancing security. By using a pen pressure-sensitive tablet, dynamic information such as the trajectory of the pen in the air can be captured, making it harder for forgers to imitate signatures [REF2]. The tablet reports whether the pen is touching the writing screen or in the air, providing additional information that is not easily available to forgers [REF2]. This integration of dynamic information into the representation of signatures adds an extra layer of security to the verification process [REF2].

Single representations are also employed in information retrieval tasks, such as passage retrieval in question answering systems. In the absence of explicit groundings to objects or actions, dialogue systems implicitly refer to an external world during conversations [REF7]. To address this, virtual embodiment has been proposed as a strategy to ground language research in perception [REF7]. By using single-player text adventure game frameworks, reinforcement learning agents can be trained with human dialogue within the game, enabling a more realistic and interactive experience [REF7].

In summary, single representations are a valuable approach within representation-focused systems. They simplify complex information by condensing it into a single entity, allowing for easier manipulation and analysis. Whether used in interactive virtual environments, natural language processing, signature verification, or information retrieval, single representations provide a means to capture and utilize essential information effectively.

References:

[REF0] - Physical actions include get, drop, put, give, steal, wear, remove, eat, drink, hug and hit, each taking either one or two arguments, e.g. put robes in closet. Every action has an explicit unambiguous effect on the underlying game state, and can only be executed if constraints are met, e.g. if the agent is holding the robes in the latter example. Emotes include applaud, blush, cringe, cry, dance, frown . . . , sulk, wave, wink (22 in total) and have no effect on the game state other than to notify nearby characters of the emote, which can have effects on their behavior. See Appendix E for further detailed descriptions.

[REF1] - Then, each candidate is scored by computing a softmax over all candidates. Unlike the BERT-based Bi-Ranker, the concatenation of the context with each individual candidate allows the model to attend to the context when encoding each candidate, building a context-dependent representation of each candidate. In contrast, the Bi-Ranker can use self-attention to build the candidate and context representations, but cannot modify their representation based upon the context. However, the Cross-Encoder is far more computationally expensive (∼11,000 slower than the Bi-Ranker for dialogue retrieval) as each concatenated representation must be recomputed, while the Bi-Ranker can cache the candidates for reuse (see Appendix B). Generative Models Similarly to the ranking setting, we use the Transformer Memory Network from Dinan et al. (2019b) to encode the context features (such as dialogue, persona, and setting). However, to predict an action, emote, or dialogue sequence, we use a Transformer architecture to decode while attending to the encoder output. Query: chicken pirate coffin rake tavern meadow objects chicken coop Pirate swords the remains shovel Ale bottles flower pot eggs dock remains garden beer fruit a pen for the chickens cargo bones a garden mug of mead An enchanted amulet.

[REF2] - 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.

[REF7] - [cs.CL] 7 Mar 2019grounded in perception. Models typically take the last few utterances from the dialogue history as input, and output a new utterance. While some goal-directed setups may use external knowledge bases (e.g. flight data for airline booking), dialogues tend to implicitly refer to an external world during the conversations without explicit grounding to objects or actions. Several position papers have proposed virtual embodiment as a strategy for language research (Brooks, 1991; Kiela et al., 2016; Gauthier and Mordatch, 2016; Mikolov et al., 2016; Lake et al., 2017). Single-player text adventure game frameworks for training reinforcement learning agents exist, i.e., Narasimhan et al. (2015) and TextWorld (Cotˆ e et al. ´ , 2018), but these do not have human dialogue within the game. Yang et al. (2017) and Bordes et al.

References given to GPT:

[REF0] - paperID: ./papers_pdf/paper_section/Representation-focused_Systems-Single_Representations/BIBREF48_f7c455cc5a40d2a31b63ac2657c9d2d6c53b1be5.pdf Title: Learning to Speak and Act in a Fantasy Text Adventure Game Chunk of text: Physical actions include get, drop, put, give, steal, wear, remove, eat, drink, hug and hit, each taking either one or two arguments, e.g. put robes in closet. Every action has an explicit unambigu ous effect on the underlying game state, and can only be executed if constraints are met, e.g. if the agent is holding the robes in the latter example. Emotes include applaud, blush, cringe, cry, dance, frown . . . , sulk, wave, wink (22 in total) and have no effect on the game state other than to notify nearby characters of the emote, which can have effects on their behavior. See Appendix E

for further detailed descriptions. Interaction Now that we have a fully realized underlying environment, we can attempt to learn and evaluate agents that can act and speak within it.

[REF1] - paperID: ./papers_pdf/paper_section/Representation-focused_Systems-Single_Representations/BIBREF48_f7c455cc5a40d2a31b63ac2657c9d2d6c53b1be5.pdf
Title: Learning to Speak and Act in a Fantasy Text Adventure Game Chunk of text: Then, each candidate is scored by computing a soft max over all candidates. Unlike the BERT-based Bi-Ranker, the concatenation of the context with each individual candidate allows the model to at tend to the context when encoding each candi date, building a context-dependent representation of each candidate. In contrast, the Bi-Ranker can use self-attention to build the candidate and con text representations, but cannot modify their rep resentation based upon the context. However, the Cross-Encoder is far more computationally expen sive (~11,000 slower than the Bi-Ranker for dia logue retrieval) as each concatenated representa tion must be recomputed, while the Bi-Ranker can cache the candidates for reuse (see Appendix B). Generative Models Similarly to the ranking set ting, we use the Transformer Memory Network from Dinan et al. (2019b) to encode the context features (such as dialogue, persona, and setting). However, to predict an action, emote, or dialogue sequence, we use a Transformer architecture to de code while attending to the encoder output. Query: chicken pirate coffin rake tavern meadow objects chicken coop Pirate swords the remains shovel Ale bottles flower pot eggs dock remains garden beer fruit a pen for the chickens cargo bones a garden mug of mead An enchanted amulet.

[REF2] - 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.

[REF3] - 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 popu lar benchmark dataset for reading comprehension. Annotators were presented with a Wikipedia para graph, and asked to write questions that could be answered from the given text. Although SQuAD has been used previously for open-domain QA re search, 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 ques tion 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

[REF4] - 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: The Siamese network has two input fields to compare two patterns and one output whose state value corresponds to the similarity between the two patterns. Two separate sub-networks based on Time Delay Neural Networks (Lang and Hinton, 1988, Guyon et al. 1990) act on each input pattern to extract features, then the cosine of the angle between two feature vectors is calculated and this represents the distance value. Results for two different subnetworks are reported here. Architecture 1 is shown in Fig 1. Architecture 2 differs in the number and size of

layers. The input is 8 by 200 units, the first convolutional layer is 6 by 192 units with each unit's receptive field covering 8 by 9 units of the input. The first averaging layer is 6 by 64 units, the second convolution layer is 4 by 57 with 6 by 8 receptive fields and the second averaging layer is 4 by 19.

[REF5] - 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) pro pose to encode candidate answer phrases as vectors and directly retrieve the answers to the input ques tions 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 dur ing 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 per formance, without relying on additional pretraining or complex joint training schemes. DPR has also been used as an important mod ule in very recent work. For instance, extending the idea of leveraging hard negatives, Xiong et al. (2020a) use the retrieval model trained in the pre vious iteration to discover new negatives and con struct a different set of examples in each training iteration.

[REF6] - paperID: ./papers_pdf/paper_section/Representation-focused_Systems-Single_Representations/BIBREF52_7b577ba0e4230b2ac58d297b3d2cfc3d2f1aaace.pdf
Title: Optimizing Dense Retrieval Model Training with Hard Negatives Chunk of text: Training hours for PQ=6 and PQ=12 are blank because they are not supported on GPU. The last line shows the performance with uncompressed index. Index Quality Train Dev Test PQ GB MRR@10 Hours MRR@10 NDCG@10 6 0.1 0.050 - 0.304 0.627 12 0.2 0.151 - 0.318 0.635 24 0.2 0.221 3.0 0.324 0.644 48 0.5 0.254 3.2 0.327 0.652 96 0.9 0.273 3.7 0.326 0.656 – 26 0.309 3.6 0.329 0.661 methods converge very fast. For example, on the passage retrieval task, ANCE needs 600k steps with batch size of 64 while ADORE needs 60k steps with batch size of 32. Secondly, to periodically update the static hard negatives, ANCE iteratively encodes the corpus to embeddings and builds temporary document indexes, which takes 10.75 hours each time with three GPUs. In contrast, STAR only builds one temporary index and ADORE does not even have this overhead. Note that although ADORE retrieves documents at each step, the search is very efficient and takes a total of 40 minutes, which is about 20% of the entire training time.

[REF7] - paperID: ./papers_pdf/paper_section/Representation-focused_Systems-Single_Representations/BIBREF48_f7c455cc5a40d2a31b63ac2657c9d2d6c53b1be5.pdf
Title: Learning to Speak and Act in a Fantasy Text Adventure Game Chunk of text: [cs.CL] 7 Mar 2019grounded in perception. Models typically take the last few utterances from the dialogue history as input, and output a new utterance. While some goal-directed setups may use external knowledge bases (e.g. flight data for airline booking), dia logues tend to implicitly refer to an external world during the conversations without explicit ground ing to objects or actions. Several position papers have proposed virtual embodiment as a strategy for language research (Brooks, 1991; Kiela et al., 2016; Gauthier and Mordatch, 2016; Mikolov et al., 2016; Lake et al., 2017). Single-player text adventure game frame works for training reinforcement learning agents exist, i.e., Narasimhan et al. (2015) and TextWorld (Cotˆ e et al. ´ , 2018), but these do not have human dialogue within the game. Yang et al. (2017) and Bordes et al.

[REF8] - paperID: ./papers_pdf/paper_section/Representation-focused_Systems-Single_Representations/BIBREF48_f7c455cc5a40d2a31b63ac2657c9d2d6c53b1be5.pdf
Title: Learning to Speak and Act in a Fantasy Text Adventure Game Chunk of text: Such is the life of a servant I suppose. How's that scepter looking? Servant: it is almost ready sire. and the crown who would you like me to take it to? Action: get scepter from small bucket King: Here just give it back. I'll have the queen find someone. Figure 1: Example dialogue from the LIGHT dataset.

[REF9] - paperID: ./papers_pdf/paper_section/Representation-focused_Systems-Single_Representations/BIBREF52_7b577ba0e4230b2ac58d297b3d2cfc3d2f1aaace.pdf
Title: Optimizing Dense Retrieval Model Training with Hard Negatives Chunk of text: as the teacher model. 7.2.3 Cascade IR. Although this paper focuses on the retrievers, we employ cascade systems for further comparison. We report the performances of the best LeToR model and the BERT model , which use BM25 as the first-stage retriever. 7.3 Implementation Details All DR models use the RoBERTabase model as the encoder. The output embedding of the "[CLS]" token is used as the representa tion of the input text. We use the inner product to compute the relevance score and adopt the Faiss library

.....

Title: Representation-focused Systems - Multiple Representations

Representation-focused systems aim to enhance the effectiveness and efficiency of information retrieval by leveraging multiple representations. These systems utilize various techniques to generate and manipulate different representations of queries and documents, allowing for more accurate matching and retrieval. In this section, we discuss the use of multiple representations in representation-focused systems, drawing inspiration from the following references: [REF0], [REF2], [REF3], [REF5], [REF6], [REF7], [REF8], and [REF9].

One common approach in representation-focused systems is the use of query augmentation, which involves adding masked tokens to the query to expand its representation [REF0]. This step allows the system to learn new terms and re-weigh existing terms based on their importance for matching the query. By utilizing BERT's representation of each token, the system can pass the contextualized output representations through a linear layer to control the dimension of the embeddings [REF0]. This dimension control is crucial for managing the space footprint of documents and can impact query execution time, especially when transferring document representations onto the GPU [REF0].

Unlike queries, documents in representation-focused systems do not typically require the addition of masked tokens [REF2]. Instead, the document encoder filters out embeddings corresponding to punctuation symbols to reduce the number of embeddings per document [REF2]. This filtering is based on the hypothesis that contextualized embeddings of punctuation are unnecessary for the system's effectiveness. The bags of embeddings for queries and documents are computed using BERT and subsequent operations, such as normalization and convolutional neural networks (CNN) [REF2].

Late interaction is a key aspect of representation-focused systems, where the relevance score between a query and a document is estimated based on their bags of contextualized embeddings [REF2]. This late interaction allows for more effective matching and retrieval, and it has been shown to be computationally cheaper compared to existing neural rankers [REF3]. ColBERT, a representation-focused system employing late interaction over BERT, has demonstrated competitive effectiveness while being significantly cheaper in terms of latency and FLOPs [REF5]. The pruning-friendly nature of the MaxSim operations in ColBERT enables efficient end-to-end retrieval from large document collections [REF3].

In the context of representation-focused systems, the training and evaluation processes are also important considerations. During training, negative examples are considered to facilitate faster training and larger batch sizes [REF4]. Evaluation speed is a crucial factor, and representation-focused systems, such as the Bi-encoder and Cross-encoder, offer efficient computation by precomputing embeddings and performing dot products between embeddings and candidates [REF4]. These systems have been evaluated against classical sparse retrieval models, attentional neural networks, and state-of-the-art results, demonstrating their effectiveness and efficiency [REF7].

Theoretical analyses and experimental studies have been conducted to understand the fidelity and dimensionality requirements of representation-focused systems [REF8]. Theoretical bounds have been derived to achieve high fidelity with respect to sparse bag-of-words models as document

length grows [REF8]. Experimental studies have also explored the ability of models to retrieve natural language documents and capture graded notions of similarity [REF8]. These studies provide insights into the practical applications and performance of representation-focused systems.

In summary, representation-focused systems leverage multiple representations to enhance the effectiveness and efficiency of information retrieval. These systems employ techniques such as query augmentation, dimension control, late interaction, and efficient training and evaluation processes. Theoretical analyses and experimental studies contribute to our understanding of the fidelity and dimensionality requirements 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 BERTChunk of text: We denote the padding with masked tokens as query augmen tation, 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).

[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: If the input se quence is the concatenation of two sentences (eg. [QUESTION ANSWER]) segment inputs of first sentence tokens are 0 and segment inputs of sec ond 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 at tention, and is computed as follows: 15% of the to kens 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 un changed (10% of the time). The masked sentence is encoded by the transformer, and the final hid den 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.

[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 BERTChunk of text: Unlike queries, we do not append [mask] tokens to documents. Af ter passing this input sequence through BERT and the subsequent linear layer, the document encoder lters out the embeddings corre sponding to punctuation symbols, determined via a pre-de ned list. is ltering is meant to reduce the number of embeddings per doc ument, 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 rel evance score of d to q, denoted as Sq,d , is estimated via late in teraction between their bags of contextualized embeddings.

[REF3] - 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 BERTChunk of text: Relative to existing neural rankers (especially, but not exclu sively, 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 con trast, 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 docu ments to be ranked is too large for exhaustive evaluation of each possible candidate document, particularly when we are only inter ested in the highest scoring ones. Concretely, we focus here on retrieving the top-k results directly from a large document collec tion 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.

[REF4] - 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 com puted 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 re trieval system, a Bi-encoder allows for the pre computation of the embeddings of all possible candidates of the system. After computing of the context embedding yctxt, the only operation re maining 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 be tween 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 con catenated into a single vector, which is encoded using one transformer.

[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 BERTChunk of text: In contrast with this trend, ColBERT (which employs late inter action over BERTbase) performs no worse than the original adap tation 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 ec tiveness, 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 mech anism, 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 mil liseconds), this di erence is explained by the time it takes to gather, stack, and transfer the document embeddings to the GPU. In partic ular, 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.

[REF6] - 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: In this setting, the context and candidate are surrounded by the special

tokens [CLS] and {sep} and con catenated into a single vector, which is encoded using one transformer. We consider the first output of the transformer as the context-candidate em bedding: yctxt,cand = h0 = f irst(ctxt, cand) where f irst is the function that takes the first vec tor of the sequence of vectors produced by the transformer. By using a single transformer, the Cross-encoder is able to perform self-attention be tween the context and candidate, resulting in the extraction of a lot of information. Scoring To score one candidate, a linear layer W is applied to the embedding yctxt,cand to reduce it from a vector to a scalar. s(ctxt, candi) = yctxt,candiW Similarly to what is done for Bi-encoder, the network is trained to minimize a cross entropy loss where the logits are s(ctxt, cand0), ..., s(ctxt, candn) where cand0 is the correct candidate and the others are nega tives taken from the training set.

[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: 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-of the-art published results where available. Our evaluations include open retrieval benchmarks (MS MARCO passage and document), and pas sage 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 state of-the-art retrieval results in MS MARCO. Our code is publicly available at 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.

[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: Our theoretical results focus on the first aspect, and de rive theoretical and empirical bounds on the suffi cient dimensionality to achieve high fidelity with respect to sparse bag-of-words models as docu ment length grows, for two types of linear random projections. The theoretical setup differs from modeling for realistic information-seeking scenar ios in at least two ways. First, trained non-linear dual encoders might be able to detect precise word overlap with much lower-dimensional encodings, especially for queries and documents with a natural distribution, which may exhibit a low-dimensional subspace structure. Second, the semantic generalization as pect of the IR task may be more important than the first aspect for practical applications, and our the ory does not make predictions about how encoder dimensionality relates to such ability to compute general semantic similarity. We relate the theoretical analysis to text re trieval in practice through experimental studies on three tasks. The first task, described in § 5, tests the ability of models to retrieve natural language documents that exactly contain a query and eval uates both BM25 and deep neural dual encoders on a task of detecting precise word overlap, de fined over texts with a natural distribution. The second task, described in § 6, is the passage re trieval sub-problem of the open-domain QA ver sion of the Natural Questions (Kwiatkowski et al., 2019; Lee et al., 2019); this benchmark reflects the need to capture graded notions of similarly and has a natural query text distribution.

[REF9] - 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: Introduction Mastering the ability to communicate with hu mans is a fundamental goal of AI. Our interaction with machines is crucial for any future application of intelligent agents in our daily lives. There are various ways for a model to deter mine what to say next in a conversation, though these methods can be distilled into two main ap proaches: generative models, which generate a ∗ Joint First Authors. sequence of text, and retrieval/ranking models, which rank candidates among a fixed set and select the optimal next utterance for a model. We focus on the latter approach in this work, as it remains superior than the former in terms of

engagingness (Shuster et al., 2018; Zhang et al., 2018a), and al lows for more control over the possible outcomes. Recently, substantial improvements to state-of the-art benchmarks on a variety of language under standing tasks have been achieved through the use of deep pretrained language models (Devlin et al., 2018); more generally, researchers have shown that by simply fine-tuning these large pretrained models, one can obtain performance gains on a number of language-related tasks. Specifically, we use the BERT models from (Devlin et al., 2018), which have been pretrained on Wikipedia and the Toronto Books Corpus (Zhu et al., 2015b). In our work, we additionally explore the pretraining of these large transformers using a different dataset that is more related to dialogue.

.....

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

Representation-focused systems play a crucial role in various domains, including web search and information retrieval. These systems aim to improve the accuracy and effectiveness of matching queries with relevant documents by focusing on the representation of both queries and documents. In this section, we will explore the concept of fine-tuning representation-focused systems, which involves optimizing the system parameters to enhance their performance [REF7].

One challenge in representation-focused systems is the large size of the term vector, which represents the bag-of-words features in information retrieval. The vocabulary size used for indexing web documents is typically very large, making the input layer of the neural network unmanageable for inference and model training. To address this issue, a method called "word hashing" has been developed, which utilizes linear hidden units in the first layer of the deep neural network [REF0]. This approach reduces the dimensionality of the input, making it more feasible for efficient inference and training.

Another important aspect of fine-tuning representation-focused systems is the evaluation of their performance. In the context of web search, relevance scores are commonly used to rank documents based on their semantic relevance to a given query. These scores are typically generated using various ranking models, such as deep structured semantic models (DSSM), topic models, and linear projection models. The performance of these models is often measured using metrics like mean Normalized Discounted Cumulative Gain (NDCG) [REF1]. Additionally, significance tests, such as the paired t-test, can be conducted to assess the statistical significance of the results [REF1].

Furthermore, the effectiveness of representation-focused systems can be enhanced by incorporating additional techniques. For example, the use of question-answer pairs can improve the performance of systems like Dense Passage Retrieval (DPR) by leveraging more supervision and achieving state-of-the-art results [REF2]. Additionally, techniques like word translation models and latent semantic models have been explored to address language discrepancies between queries and documents, improving the semantic matching process [REF5] [REF6].

To optimize the performance of representation-focused systems, the fine-tuning process involves adjusting the model parameters. This can be achieved through supervised training methods, where the model parameters, such as weight matrices and bias vectors, are learned to maximize the likelihood of relevant documents given queries [REF4]. The fine-tuning process also involves minimizing the cross-entropy error between the original term vector and the reconstructed term vector, ensuring that the model parameters are optimized for differentiating relevant documents from irrelevant ones [REF7].

In conclusion, fine-tuning representation-focused systems is a crucial step in improving their performance. By addressing challenges related to input dimensionality, evaluating performance using appropriate metrics, and incorporating additional techniques, these systems can be optimized to enhance the accuracy and effectiveness of matching queries with relevant documents.

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/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 hyper parameters 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.

[REF2] - 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: Al though the results of DPR on WQ and TREC in the single-dataset setting are less competitive, adding more question–answer pairs helps boost the perfor mance, achieving the new state of the art. To compare our pipeline training approach with joint learning, we run an ablation on Natural Ques tions 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 supervi sion, while outperforming a comparable joint train ing approach with a simpler design (Appendix D). One thing worth noticing is that our reader does consider more passages compared to ORQA, al though 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 sin gle 32GB GPU, thus the latency remains almost identical to the single passage case (around 20ms). The exact impact on throughput is harder to mea sure: ORQA uses 2-3x longer passages compared to DPR (288 word pieces compared to our 100 tokens) and the computational complexity is super linear in passage length. We also note that we found k = 50 to be optimal for NQ, and k

[REF3] - 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 high dimensional 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.

[REF4] - 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.

[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: INTRODUCTION Modern search engines retrieve Web documents mainly by matching keywords in documents with those in search queries. However, lexical matching can be inaccurate due to the fact that a concept is often expressed using different vocabularies and language styles in documents and queries. Latent semantic models such as latent semantic analysis (LSA) are able to map a query to its relevant documents at the semantic level where lexical matching often fails (e.g.,). These latent semantic models address the language discrepancy between Web documents and search queries by grouping different terms that occur in a similar context into the same semantic cluster. Thus, a query and a document, represented as two vectors in the lower-dimensional semantic space, can still have a high similarity score even if they do not share any term. Extending from LSA, probabilistic topic models such as probabilistic LSA (PLSA) and Latent Dirichlet Allocation (LDA) have also been proposed for semantic matching . However, these models are often trained in an unsupervised manner using an objective function that is only loosely coupled with the evaluation metric for the retrieval task.

[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 second is a word translation model (WTM in Row 3) which is intended to directly address the query-document language discrepancy problem by learning a lexical mapping between query words and document words . The third includes a set of state-of-the-art latent semantic models which are learned either on documents only in an unsupervised manner (LSA, PLSA, DAE as in Rows 4 to 6) or on clickthrough data in a supervised way (BLTM-PR, DPM, as in Rows 7 and 8). In order to make the results comparable, we re implement these models following the descriptions in , e.g., models of LSA and DPM are trained using a 40k-word vocabulary due to the model complexity constraint, and the other models are trained using a 500K-word vocabulary. Details are elaborated in the following paragraphs. TF-IDF (Row 1) is the baseline model, where both documents and queries represented as term vectors with TF-IDF term weighting. The documents are ranked by the cosine similarity between the query and document vectors. We also use BM25 (Row 2) ranking model as one of our baselines.

[REF7] - 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, the model parameters are fine tuned 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. Their evaluation shows that the SH approach achieves a superior document retrieval performance to the LSA. However, SH suffers from two problems, and cannot outperform the standard lexical matching based retrieval model (e.g., cosine similarity using TF-IDF term weighting). The first problem is that the model parameters are optimized for the re-construction of the document term vectors rather than for differentiating the relevant documents from the irrelevant ones for a given query. 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.

[REF8] - 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 pre trained models, along with cross-attention mecha nisms, 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 re trieve relevant passages iteratively using reformu lated question vectors. As an alternative approach that skips passage retrieval, Seo et al. (2019) pro pose to encode candidate answer phrases as vectors and directly retrieve the answers to the input ques tions 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.

[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: 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 fine tuned 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.

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

In the field of information retrieval, retrieval architectures play a crucial role in efficiently searching for relevant information. One important problem in retrieval architectures is the Maximum Inner Product Search (MIPS) [REF2]. MIPS involves finding a data vector from a collection of "database" vectors that maximizes the inner product with a given query vector [REF2]. This problem arises in various applications such as recommendation systems, multi-class

prediction, and vision problems [REF2]. To address the MIPS problem, Shrivastava and Li (2014a) propose constructing a Locality Sensitive Hash (LSH) for inner product similarity [REF2]. LSH is a popular tool for approximate nearest neighbor search and has been widely used in different settings [REF2].

One approach to LSH for inner product similarity is the L2-ALSH(SL) method, which is parameterized by m, U, and r [REF1]. The L2-ALSH(SL) method utilizes a pair of mappings, P(x) and Q(y), combined with the standard L2 hash function to generate the hash values [REF1]. However, L2-ALSH(SL) is not universal and requires tuning parameters specific to the threshold S and ratio c [REF8]. In contrast, SIMPLE-LSH, another LSH method, does not require any parameters and is both symmetric and universal [REF4]. Empirical comparisons have shown that SIMPLE-LSH outperforms L2-ALSH(SL) in terms of hashing quality [REF4].

The optimization of LSH parameters for the best hashing quality is a non-convex optimization problem [REF0]. Shrivastava and Li (2014a) suggest using grid search to find a bound on the optimal hashing quality ρ [REF0]. By optimizing over the parameters m, U, and r, the best ρ can be obtained for a given threshold S and ratio c [REF0]. It is important to note that L2-ALSH(SL) is not an (S, cS)-ALSH for all choices of S and c, making it less desirable for MIPS problems where the threshold S can vary with the query [REF8].

In addition to L2-ALSH(SL) and SIMPLE-LSH, there are other variations of LSH methods for inner product similarity, such as SIGN-ALSH(SL) [REF4]. SIGN-ALSH(SL) is based on random projections and incorporates an asymmetric transform similar to L2-ALSH(SL) [REF4]. However, it is worth noting that asymmetric LSH is not necessary for the MIPS setting when queries are normalized and database vectors are bounded [REF3]. In this case, a universal symmetric LSH is possible [REF3].

In conclusion, retrieval architectures and vector search algorithms play a crucial role in efficiently searching for relevant information. The MIPS problem, in particular, has been addressed using LSH methods such as L2-ALSH(SL), SIMPLE-LSH, and SIGN-ALSH(SL). While L2-ALSH(SL) requires tuning parameters and is not universal, SIMPLE-LSH is parameter-free and outperforms L2-ALSH(SL) in terms of hashing quality. The choice of LSH method depends on the specific requirements of the problem, such as the normalization of queries and the boundedness of database vectors.

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 L2-ALSH(SL) and SIGN-ALSH(SL), for each desired threshold S and ratio c, one can optimize over the parame ters 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.

[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 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 vari able on [0, r]. The alphabet Γ used is the integers, the in termediate 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))).

[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: 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 us ing 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 so lutions, Shrivastava and Li (2014a) suggest constructing a Locality Sensitive Hash (LSH) for inner product "similar ity". Proceedings of the 31 st International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copy right 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).

[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 mo tivate 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 advan tage to asymmetry. In the third setting, where an asym metric hash is indeed needed, the hashes suggested by Shrivastava and Li (2014a;b) are not ALSH, and a differ ent asymmetric hash is required (which we provide). Fur thermore, even in the MIPS setting when queries are nor malized (the second setting), the asymmetric hashes sug gested 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 uni versal. 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 pa rameters. As can be seen in the Figures, SIMPLE-LSH shows a dra matic empirical improvement over L2-ALSH(SL). Follow ing 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 ran dom projections, as is SIMPLE-LSH, but with an asym metric 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: The only asym metry allowed is in the problem definition, as we allow re quiring the property for differently constrained x and y. This should be contrasted with a truly asymmetric hash, where two different functions are used, one for each space. Formally, an asymmetric hash for a pair of spaces X and Y is a joint distribution over pairs of mappings (f, g), f : X → Γ, g : Y → Γ. The asymmetric hashes we con sider will be specified by a pair of deterministic mappings P : X → Z and Q : Y → Z and a single random map ping (i.e. distribution over functions) h : Z → Γ, where f(x) =

h(P(x)) and g(y) = h(Q(y)). Given a similarity function sim : X × Y → R we define: Definition 2 (Asymmetric Locality Sensitive Hash ing (ALSH)).

[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 mea sures such as set containment and weighted Jaccard simi larity. 6. Conclusion We provide a complete characterization of when symmetric and asymmetric LSH are possible for inner product similar ity: • Over R d , no symmetric nor asymmetric LSH is possi ble. • 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: [f(xi) = g(xj)]. Setting θ = (p1 + p2)/2 < 1 and ϱ = (p1 − p2)/2 > 0, the ALSH property implies that for every i, j: Z(i, j)(P(i, j) − θ) ≥ ϱ (3) or equivalently: Z ⊙ P − θ ϱ ≥ 1 (4) where ⊙ denotes element-wise (Hadamard) product. Now, for a sign matrix Z, the margin complexity of Z is defined as mc(Z) = infZ⊙X≥1 kXkmax (see Srebro and Shraibman, 2005, and also for the definition of the max-norm kXkmax), and we know that the margin complexity of an N × N triangular matrix is bounded by mc(Z) = Ω(log N) (Forster et al., 2003), implying k(P − θ)/ϱkmax = Ω(log N).

[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: They furthermore calculate the hashing quality ρ as a func tion of m, U and r, and numerically find the optimal ρ over a grid of possible values for m, U and r, for each choice of S, c. Before moving on to presenting a symmetric hash for the problem, we note that L2-ALSH(SL) is not universal (as de fined at the end of Section 2). That is, not only might the optimal m, U and r depend on S, c, but in fact there is no choice of the parameters m and U that yields an ALSH for all S, c, or even for all ratios c for some specific threshold S or for all thresholds S for some specific ratio c. This is unfortunate, since in MIPS problems, the relevant thresh old S is the maximal inner product maxx∈S q ⊤x (or the threshold inner product if we are interested in the "top-k" hits), which typically varies with the query. It is therefore desirable to have a single hash that works for all thresholds. Lemma 1. For any m, U, r, and for any 0 < S < 1 and 1 − U 2m+1−1 (1 − S 2m+1) 2S ≤ c < 1, L2-ALSH(SL) is not an (S, cS)-ALSH for inner product similarity over X• = {x|kxk ≤ 1} and Y ◦ = {q|kqk = 1}.

[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: But when queries and data vectors are bounded and queries are not normalized, we do observe the power of asymmetry: here, a symmetric LSH is not possible, but an asymmetric LSH exists (Section 5). As mentioned above, our study also yields an LSH for MIPS, which we refer to as SIMPLE-LSH, which is not only symmetric but also parameter-free and enjoys significantly better theoretical and empirical compared to L2-ALSH(SL) proposed by Shrivastava and Li (2014a). In Appendix A we show that all of our theoretical observations about L2- ALSH(SL) apply also to the alternative hash SIGN-LSH(SL) put forth by Shrivastava and Li (2014b). The transformation at the root of SIMPLE-LSH was also re cently proposed by Bachrach et al. (2014), who used it in a PCA-Tree data structure for speeding up the Xbox rec ommender system. Here, we study the transformation as part of an LSH scheme, investigate its theoretical proper ties, and compare it to LS-ALSH(SL). 2.

.....

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

Locality sensitive hashing (LSH) is a popular technique used in retrieval architectures and vector search to efficiently search for similar objects in high-dimensional spaces. LSH works by hashing nearby objects into the same bucket, allowing for fast retrieval of candidate sets for similarity search [REF9]. However, traditional LSH methods often require a large number of hash tables to achieve high search accuracy, leading to space inefficiency [REF1]. In this section, we will explore different LSH approaches, particularly focusing on locality sensitive hashing schemes based on p-stable distributions and multi-probe LSH.

One approach to improve the space efficiency of LSH is to use p-stable distributions. These distributions, based on stable distributions, are defined as limits of normalized sums of independent identically distributed variables [REF0]. By utilizing p-stable distributions, LSH can be designed to work for all values of p in the range (0, 2) [REF0]. This approach allows for the reduction of space requirements while maintaining search accuracy, making it a promising solution for high-dimensional datasets [REF1].

Another approach to enhance the efficiency of LSH is the multi-probe LSH method. Traditional LSH methods use a single hash table to generate candidate sets, but multi-probe LSH utilizes multiple probes to further refine the candidate set [REF5]. By probing multiple hash tables, multi-probe LSH can achieve similar search quality with fewer hash tables, thus reducing the space requirement [REF5]. This method trades off time for space, as it requires additional query time to perform the multiple probes [REF1]. However, experimental studies have shown that multi-probe LSH can significantly improve space efficiency while maintaining search accuracy [REF1].

To implement multi-probe LSH, a sequence of perturbation vectors is designed, where each vector maps to a unique set of hash values [REF3]. This ensures that each hash bucket is probed only once, avoiding duplicate computations [REF3]. The perturbation vectors are precomputed, reducing the query time overhead of dynamically generating them at query time [REF2]. By applying these perturbation vectors to the hash values of the query object, the multi-probe LSH method avoids the computational overhead associated with point perturbation and hash value computations [REF3].

In summary, retrieval architectures and vector search can benefit from locality sensitive hashing approaches such as p-stable distributions and multi-probe LSH. These methods offer space-efficient solutions for high-dimensional datasets while maintaining search accuracy. By utilizing p-stable distributions, LSH can be designed to work for a wide range of values of p. On the other hand, multi-probe LSH reduces space requirements by probing multiple hash tables and utilizing precomputed perturbation vectors. These approaches provide efficient retrieval architectures for vector search applications in various domains.

References given to GPT:
[REF0] - 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 dis tributions, that works for all p 2 (0; 2Ȼ. 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 defini tion follows).

[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: To achieve high search accuracy, the LSH method needs to use multiple hash ta bles 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 space efficiency requirement. In a recent theoretical study , Panigrahy proposed an entropy-based LSH method that generates randomly “per turbed” 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 entropy based method can reduce the space requirement of the basic LSH method, significant improvements are possible.

[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: As we will explain shortly, it turns out that we know the distribution of the zj values precisely and can compute E[z 2 j] for each j. This motivates the following optimization: We approximate the z 2 j values by their expectations. Using this approximation, the sorted order of perturbation sets can be precomputed (since the score of a set is a function of the z 2 j values). The generation process is exactly the same as described in the previous subsection, but uses the E[z 2 j] val ues instead of their actual values. This can be done inde pendently of the query q. At query time, we compute the mapping π t j as a function of query q (separately for each hash table t). These mappings are used to convert each per turbation set in the precomputed order into L perturbation vectors, one for each of the L hash tables. This precom putation reduces the query time overhead of dynamically generating the perturbation sets at query time.

[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: Recall that the LSH functions we use are of the form ha,b(v) = b a·v+b W c. If we pick W to be reasonably large, with high probability, similar objects should hash to the same or adjacent values (i.e. differ by at most 1). Hence we restrict our attention to perturbation vectors Δ with δi ∈ {−1, 0, 1}. Each perturbation vector is directly applied to the hash values of the query object, thus avoiding the overhead of point perturbation and hash value computations associated with the entropy-based LSH method. We will design a se quence of perturbation vectors such that each vector in this sequence maps to a unique set of hash values so that we never probe a hash bucket more than once. Figure 1 shows how the multi-probe LSH method works. In the figure, gi(q) is the hash value of query q in the i-th table, (Δ1, Δ2, . . .)

[REF4] - 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 re sults 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.

[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: However, for a given K, the multi-probe LSH method can effectively re duce 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 dif ferent LSH methods (C is candidate set size, N is dataset size): multi-probe LSH has similar filter ra tio as other LSH methods. The similarity search problem is closely related to the nearest neighbor search problem, which has been studied ex tensively.

[REF6] - 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: 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 dif ferent LSH methods (C is candidate set size, N is dataset size): multi-probe LSH has similar filter ra tio as other LSH methods. The similarity search problem is closely related to the nearest neighbor search problem, which has been studied ex tensively. A number of indexing data structures have been devised for nearest neighbor search; examples include R tree , K-D tree , and SR-tree . These data struc tures are capable of supporting similarity queries, but do not scale satisfactorily to large, high-dimensional datasets. The exact nearest neighbor problem suffers from the “curse of dimensionality” – i.e. either the search time or the search space is exponential in the number of dimensions, d

[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: For each word segment, we then use the Marsyas library to extract feature vectors by taking a 512-sample sliding win dow with variable stride to obtain 32 windows for each word. For each of the 32 windows, we extract the first six MFCC parameters, resulting in a 192-dimensional feature vector for each word. Table 1 summarizes the number of objects in each datasetDataset #Objects #Dimension Total Size Image 1,312,581 64 336 MB Audio 2,663,040 192 2.0 GB Table 1: Evaluation Datasets. and the dimensionality of the feature vectors. 5.2 Evaluation Benchmarks For each dataset, we created an evaluation benchmark by randomly picking 100 objects as the query objects, and for each query object, the ground truth (i.e., the ideal answer) is defined to be the query object’s K nearest neighbors (not including the query object itself), based on the Euclidean distance of their feature vectors. Unless otherwise specified, K is 20 in our experiments.

[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: However, per forming this calculation is cumbersome. Instead, Panigrahy proposes a clever way to sample buckets from the distri bution 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 sam pling 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.

[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: It has been shown in that when the dimensionality exceeds about 10, existing index ing data structures based on space partitioning are slower than the brute-force, linear-scan approach. For high-dimensional similarity search, the best-known in dexing method is locality sensitive hashing (LSH) . The basic method uses a family of locality-sensitive hash func tions to hash nearby objects in the high-dimensional space into the same bucket. To perform a similarity search, the in dexing method hashes a query object into a bucket, uses the data objects in the bucket as the candidate set of the results, and then ranks the candidate objects using the

distance measure of the similarity search. To achieve high search accuracy, the LSH method needs to use multiple hash ta bles 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.

.....

Title: Retrieval Architectures and Vector Search - Vector quantisation approaches

Vector quantisation is a widely used technique in retrieval architectures and vector search. It involves mapping high-dimensional vectors to a set of centroids in a codebook [REF8]. This section explores different vector quantisation approaches, focusing on product quantizers, variable-rate vector quantizers, and hierarchical quantizers.

Product quantizers are scalar quantizers where each component has a different quantization function [REF0]. They are known for their ability to produce a large set of centroids from several small sets of centroids associated with subquantizers. The learning process of product quantizers involves using a limited number of vectors and adapting the codebook to represent the data distribution [REF0]. However, the explicit storage of the codebook can be inefficient [REF0].

Variable-rate vector quantizers leverage the similarities between variable-rate vector quantizers and decision trees for statistical pattern classification [REF1]. These quantizers can be used in conjunction with entropy coding to achieve further compression, but at the expense of added complexity and variable-rate coding [REF1]. Designing vector quantizers specifically for entropy-constrained applications has shown excellent compression performance [REF1].

Hierarchical quantizers have gained attention due to their ability to efficiently assign descriptors to a large number of centroids [REF9]. They address the limitations of traditional quantizers, such as the large number of samples required for learning and the prohibitive complexity of the algorithm itself [REF6]. Hierarchical quantizers, such as hierarchical k-means (HKM), improve the efficiency of the learning stage and the assignment procedure [REF6]. However, memory usage and the size of the learning set remain challenges [REF6].

Inverted file with asymmetric distance computation (IVFADC) indexing systems have been proposed to address the problem of efficient vector search [REF3]. Multiple assignment strategies are used to assign a query vector to multiple indexes, corresponding to the nearest neighbors in the codebook [REF3]. However, applying multiple assignment to database vectors can increase memory usage [REF3].

Recent advancements in vector quantisation have focused on limiting memory usage, which is crucial for problems involving large amounts of data [REF4]. Methods such as global GIST descriptor mapping and spectral hashing (SH) have been developed to reduce memory requirements while maintaining search efficiency [REF4]. These techniques approximate the search of Euclidean nearest neighbors by searching for nearest neighbors in terms of Hamming distances between codes [REF4].

In conclusion, vector quantisation approaches play a vital role in retrieval architectures and vector search. Product quantizers, variable-rate vector quantizers, hierarchical quantizers, and inverted file systems with asymmetric distance computation are among the techniques used to efficiently map high-dimensional vectors to centroids. Recent advancements have focused on reducing memory usage while maintaining search efficiency. These approaches offer valuable solutions for various applications, including image and voice coding, scene recognition, and large-scale data indexing.

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: 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∗$\square\square\square\square\square$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.

[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 com pression 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 impor tant 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 vec tor 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: 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 serendipi tous structural properties of the codes designed by iterative clustering algo rithms. As an example of the desirable struct ural properties of vector quan tizers, 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.

[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: 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 correspond ing inverted lists are scanned. Multiple assignment is not applied to database vectors, as this would increase the memory usage.

[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: 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 be tween codes. In , spectral hashing (SH) is shown to outperform the binary codes generated by the restricted Boltzmann machine , boosting and LSH.

[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: Chapter 5 treats the basics of simple scalar quantization: the performance charac teristics and common high resolution approximations developed by Ben nett. Chapter 6 describes the optimality properties of simple quantizers, the structure of high-resolution optimal quantizers, and the basic design algorithm used throughout the book to design codebooks, the algorithm developed by Stuart Lloyd of Bell Laboratories in the mid 1950s. Chapters 7 and 8 build on scalar quantizers by operating on the signal before quanti zation so as to make the quantization more efficient. Such pre-processing is intended to remove some of the redundancy in the signal, to reduce the sig nal variance, or to concentrate the signal energy. All of these properties can result in better performance for a given bit rate and complexity if properly used. Chapter 7 concentrates on predictive quantization wherein a linear prediction based on past reconstructed values is removed from the signal and the resulting prediction residual is quantized. In Chapter 8 vectors or blocks of input symbols are transformed by a simple linear and orthogo-XVlll PREFACE nal transform and the resulting transform coefficients are quantized.

[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: First, the number of samples required to learn the quantizer is huge, i.e., several times k. Second, the complexity of the algorithm itself is prohibitive. Finally, the amount of computer memory available on Earth is not sufficient to store the floating point values representing the centroids. The hierarchical k-means see (HKM) improves the efficiency of the learning stage and of the corresponding assignment procedure . However, the aforementioned limitations still apply, in particular with respect to mem ory usage and size of the learning set. Another possibility are scalar quantizers, but they offer poor quantization er ror properties in terms of the trade-off between memory and reconstruction error. Lattice quantizers offer better quantization properties for uniform vector distributions, but this condition is rarely satisfied by real world vectors. In practice, these quantizers perform significantly worse than k-means in indexing tasks .

[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: 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 dis tance 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.

[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 pur pose 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 D dimensional 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: 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 quan tizer 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 Fig ure 11. The cost of the extra quantization step required by IVFADC appears clearly for small database sizes.

.....

Title: Retrieval Architectures and Vector Search - Graph approaches

Graph-based approaches have gained significant attention in retrieval architectures and vector search due to their ability to capture complex relationships and provide efficient search capabilities. In this section, we discuss various graph approaches that have been proposed in the literature for retrieval architectures and vector search.

One approach is the use of navigable small world graphs, which exhibit logarithmic scalability in the greedy search algorithm [REF1]. These graphs, also known as navigable small world networks, have been shown to have the small world navigation property, allowing for efficient approximate k-nearest neighbor search [REF1]. The construction of navigable small world graphs can be achieved using a simple algorithm that does not require prior knowledge of the internal structure of the metric space [REF1].

Another graph-based approach is the decentralized algorithm, which utilizes lattice distance to determine the next contact in the search process [REF2]. This algorithm operates in phases, with each phase reducing the lattice distance from the current node to the target [REF2]. By selecting contacts that are closer to the target, the decentralized algorithm efficiently navigates the graph to find the desired results [REF2].

Efficient methods for constructing k-nearest neighbor graphs (K-NNG) in general metric spaces have also been proposed [REF3]. These methods aim to minimize empirical complexity while considering the specific characteristics of the distance metric or similarity measure used [REF3]. Additionally, tree-based data structures have been designed for both general metric spaces and Euclidean spaces to facilitate K-NN search [REF3].

To address the challenges posed by high-dimensional metric spaces, approximate neighbor search techniques have been developed [REF4]. These techniques aim to reduce the computational requirements while still providing accurate results [REF4]. One commonly used approach is the approximate k-nearest neighbor search, where the distance between the query and the elements in the result set is within a predefined accuracy threshold [REF4].

In terms of performance evaluation, several studies have been conducted to assess the effectiveness of different retrieval architectures and vector search approaches [REF7]. These studies consider factors such as recall, search time, and the impact of intrinsic dimensionality on performance [REF7]. Experimental results have shown that graph-based approaches can achieve high recall rates, even with faster search settings [REF7].

In conclusion, graph-based approaches offer promising solutions for retrieval architectures and vector search. Navigable small world graphs, decentralized algorithms, and efficient construction methods for K-NNGs are among the approaches that have been proposed. Additionally, approximate neighbor search techniques have been developed to address the challenges of high-dimensional spaces. Experimental evaluations have demonstrated the effectiveness of these approaches in achieving high recall rates and efficient search capabilities.

References given to GPT:
[REF0] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Graph_approaches/BIBREF69_f17c6e164ccc7ec1ad91b3fbbafe8f84664e9803.pdf Title: Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures Chunk of text: When minor loss in recall is acceptable, the fast setting reduces scan rate by nearly half. Shape has a particularly high scan rate due to its small size. 3.4 System Environment We used commodity servers of the following configuration: dual quad core Intel E5430 2.66GHz CPU; 16GB main mem ory. All machines ran CentOS 5.3 with Linux kernel 2.6.18 and gcc 4.3.4. We use OpenMP based parallelization for our own code and LSHKIT 2 . The Recursive Lanczos Bisection code 3 is not parallelized and we disabled the parallelization of our code when comparing against it. 4. EXPERIMENTAL RESULTS This section reports experimental results.

[REF1] - 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 struc ture 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.

[REF2] - 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 long range 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.

[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 con struction in general metric spaces with low empirical com plexity, 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 par titioning and space filling curves , but they do not naturally generalize to other distance metrics or general sim ilarity 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 struc tures are designed for both general metric space and Eu clidean space

[REF4] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Graph_approaches/BIBREF70_c197ecb6a6987667cadcb498136989af1827cce0.pdf Title: Approximate Nearest Neighbor Algorithm based on Navigable Small World 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 pro posed 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.

[REF5] - 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, nor malization 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)

[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: However, still there are several ways to optimize the structure in order to get lower complexity and/or better accuracy constants, such as More sophisticated algorithms for node friends selec tion (see Section 5). It is quite evident that selecting nearest neighbors as friends is not the best way to approximate Delaunay graph, since this approach takes into account only distances between the new element and candidates, and disregards distances between the candidates. Knowledge of internal structure of the metric space can boost search performance. In is was shown that for Euclidean space the accuracy of a single search can be significantly increased while keeping the number of friends per node fixed. More sophisticated algorithms for navigable small world creation.

[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 pos sible 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: 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 reli ance 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 .

[REF9] - 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: 5 the average fraction of visited elements within a single 10-NN-search with 0.999 recall for about 22 million elements dataset versus dimensionality. A plateau with an “optimal” value of the dimensionality is clearly seen from the plot. The position of the “optimum” value of dimen sionality slowly shifts with increasing dataset size, which may be attributed to shorter greedy paths at higher dimen sionality (see Section 6.2). Fig. 3. Average fraction of visited elements within a single 10-NN-search with 0.999 recall versus the size of the dataset for different dimensionality. Fig. 4. Distance calculations and the value of m to get a 0.999 recall versus the size of the dataset for d¼20.

.....

Title: Retrieval Architectures and Vector Search - Optimisations

Retrieval architectures and vector search play a crucial role in information retrieval systems, enabling efficient and accurate retrieval of relevant documents or data points. In this section, we discuss various optimisations that can be applied to retrieval architectures and vector search techniques to enhance their performance and effectiveness.

One important optimisation technique is the use of compressed encodings for vectors. In the work by [REF0], different byte encodings (16, 32, and 64 bytes) are explored for each vector. By pre-processing the vectors using techniques like Optimized Product Quantization (OPQ), the dimensionality of the vectors can be reduced, leading to more efficient storage and retrieval. The authors also vary the trade-off parameter τ to find the optimal balance between efficiency and quality.

Another approach to optimising retrieval architectures is the use of reranking models. In [REF2], a passage selection model is proposed, which utilizes cross-attention between the question and the passage. While cross-attention is not feasible for large-scale retrieval due to its non-decomposable nature, it has shown promising results in selecting passages from a small number of retrieved candidates. By leveraging BERT representations and probability calculations, the model effectively ranks passages based on their relevance to the query.

Efficient computation is another key aspect of optimising retrieval architectures. In [REF3], a batched dot-product computation is proposed, where the documents are padded to their maximum length and the dot-product is computed between the query and each document. The computation is performed on the GPU, and the scores of each document are obtained by reducing the matrix across document terms via max-pooling and across query terms via summation. This approach is shown to be computationally efficient compared to existing neural rankers.

In the context of vector search, optimisations can be applied to improve the efficiency of searching and retrieval. In [REF4], a kernel-based approach is presented, where the kernel scans the closest inverted lists for each query and calculates per-vector pair distances using lookup tables. The lookup tables are stored in shared memory, and the approach allows for efficient processing of query against inverted list pairs. Multi-pass kernels are also explored to process query against inverted list pairs independently, further improving efficiency.

Dense retrieval is another area where optimisations can be applied. In [REF5], several training methods for dense retrieval baselines are discussed, including random negative sampling and in-batch negative sampling. These methods aim to improve the efficiency and effectiveness of dense retrieval by sampling negatives from the entire corpus or using other queries' relevant documents as negative examples.

Parallel processing and hardware optimisations are also important considerations for retrieval architectures. In [REF6], the concept of warps and blocks in GPU architectures is discussed. Warps represent separate CPU hardware threads, and blocks comprise a collection of warps. Utilizing the shared memory and parallel processing capabilities of GPUs can significantly improve the efficiency and performance of retrieval architectures.

In conclusion, retrieval architectures and vector search can be optimised through various techniques such as compressed encodings, reranking models, efficient computation, kernel-based approaches, dense retrieval methods, and hardware optimisations. These optimisations aim to enhance the efficiency, effectiveness, and scalability of retrieval systems, enabling faster and more accurate retrieval of relevant information.

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 trade offs 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: 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 annota tors 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.

[REF2] - 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 cross attention between the question and the passage. Al though cross-attention is not feasible for retrieving relevant passages in a large corpus due to its non decomposable nature, it has more capacity than the dual-encoder model sim(q, p) as in Eq. (1). Apply ing 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) rep resentation 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

[REF3] - 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 docu ment, 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. Fi nally, we sort the k documents by their total scores. Relative to existing neural rankers (especially, but not exclu sively, 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.

[REF4] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Optimisations/BIBREF73_2cbb8de53759e75411bc528518947a3094fbce3a.pdf Title: Billion-Scale Similarity Search with GPUs Chunk of text: A kernel is responsible for scanning the τ closest inverted lists for each query, and calculating the per vector pair distances using the lookup tables Ti. The Ti are stored in shared memory: up to nq×τ ×maxi |Ii|×b lookups are required for a query set (trillions of accesses in practice), and are random access. This limits b to at most 48 (32-bit floating point) or 96 (16-bit floating point) with current architectures. In case we do not use the decomposition of Equation (11), the Ti are calculated by a separate kernel before scanning. Multi-pass kernels. Each nq × τ pairs of query against inverted list can be processed independently. At one ex treme, a block is dedicated to each of these, resulting in up to nq × τ × maxi |Ii| partial results being written back to global memory, which is then k-selected to nq × k final re sults.

[REF5] - 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 ran domly samples negatives from the entire corpus, and the latter uses other queries' relevant documents in the same batch as negative documents.

[REF6] - paperID: ./papers_pdf/paper_section/Retrieval_Architectures_and_Vector_Search-Optimisations/BIBREF73_2cbb8de53759e75411bc528518947a3094fbce3a.pdf Title: Billion-Scale Similarity Search with GPUs Chunk of text: Despite the "thread" terminology, the best analogy to modern vectorized multicore CPUs is that each warp is a separate CPU hardware thread, as the warp shares an instruction counter. Warp lanes taking different execu tion paths results in warp divergence, reducing performance. Each lane has up to 255 32-bit registers in a shared register file. The CPU analogy is that there are up to 255 vector registers of width 32, with warp lanes as SIMD vector lanes. Collections of warps. A user-configurable collection of 1 to 32 warps comprises a block or a co-operative thread ar ray (CTA). Each block has a high speed shared memory, up to 48 KiB in size.

[REF7] - 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.

[REF8] - 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 place this token right a□er BERT's sequence start token [CLS]. If the query has fewer than a pre-de□ned number of tokens Nq, we pad it with BERT's special [mask] tokens up to length Nq (otherwise, we truncate it to the □rst Nq tokens). □is padded sequence of input tokens is then passed into BERT's deep transformer architecture, which computes a contextualized representation of each token. We denote the padding with masked tokens as query augmen tation, 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.

[REF9] - 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: In contrast with this trend, ColBERT (which employs late inter action over BERTbase) performs no worse than the original adap tation 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□ec tiveness, 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 mech anism, 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 mil liseconds), this di□erence is explained by the time it takes to gather, stack, and transfer the document embeddings to the GPU. In partic ular, 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.

.....

Title: Learned Sparse Retrieval - Document expansion learning

In the field of information retrieval, document expansion learning has gained significant attention due to its potential to improve retrieval performance by expanding the original document representation. One approach to document expansion learning is learned sparse retrieval, which aims to leverage the regularity of recall values for recurring terms to enhance recall prediction [REF0]. By reusing recall values from previous occurrences of the same term, the need for feature generation and prediction stages can be avoided, leading to more efficient retrieval processes.

Transfer learning plays a crucial role in learned sparse retrieval, as it enables the learning process to occur across different queries [REF0]. Traditionally, retrieval tasks involve classifying documents as relevant or non-relevant to a given query. However, with transfer learning, the focus shifts towards treating retrieval as a transfer learning task, where knowledge learned from one query can be applied to improve retrieval for other queries [REF0].

To implement learned sparse retrieval, various techniques have been explored. One such technique is gradual unfreezing, which involves fine-tuning different layers of the model over time [REF1]. This approach, originally applied to language models, can be adapted to encoder-decoder models by gradually unfreezing layers in both the encoder and decoder, starting from the top [REF1]. Additionally, shared parameters, such as input embedding and output classification matrices, are updated throughout the fine-tuning process [REF1].

Support vector regression (SVR) with the radial basis function (RBF) kernel has been found to be effective in learned sparse retrieval [REF2]. SVR maps original features into a higher dimensional space and aims to find a linear solution that is as flat as possible in this space [REF2]. The RBF kernel, in particular, measures vector similarity based on the Gaussian distribution function, allowing for the fitting of complex non-linear regression models [REF2]. In the context of learned sparse retrieval, all features are treated equally by the final regression model, except for the scaling that occurs during preprocessing [REF2].

While learned sparse retrieval shows promise in improving retrieval performance, there are still theoretical inconsistencies between multiple pocket document models and single pocket relevance models that need to be addressed [REF3]. Further experiments and studies are required to gain a better understanding of the effectiveness of recall-based term weighting and its impact on retrieval models [REF3].

In the broader context of information retrieval, term weight prediction methods and semantic analysis techniques have also been explored to address term mismatch and improve retrieval performance [REF5]. These methods aim to enhance the matching of terms between documents and queries, either by identifying concepts in queries or finding synonyms for query terms [REF5]. Additionally, diagnostic interventions have been proposed to improve problematic areas of queries [REF5].

Recent advancements in sparse representation learning have shown promising results in improving the ranking performance of term-based representations while maintaining the interpretability and efficiency of bag-of-words (BoW) methods [REF6]. SparTerm, a sparse model based on pre-trained language models (PLMs), has demonstrated superior performance compared to other sparse models, even outperforming models based on larger PLMs [REF6]. This research also provides insights into how deep knowledge from PLMs can be transferred to sparse representation learning [REF6].

In conclusion, learned sparse retrieval and document expansion learning have emerged as promising approaches to enhance retrieval performance. By leveraging the regularity of recall values and applying transfer learning techniques, these methods aim to improve recall prediction and expand document representations. Techniques such as gradual unfreezing and support vector regression with the RBF kernel have shown effectiveness in learned sparse retrieval. However, further research is needed to address theoretical inconsistencies and explore the potential of term weight prediction methods and semantic analysis techniques in improving retrieval models. Additionally, advancements in sparse representation learning offer new possibilities for enhancing term-based representations in information retrieval systems.

References given to GPT:
[REF0] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Document_expansion_learning/BIBREF79_1225eb6570ce8d45067329fafcc8ff7636a65923.pdf
Title: Modeling and Solving Term Mismatch for Full-Text Retrieval Chunk of text: 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 appli cation 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 doc ument classification task, where each document needs to be classified as relevant or non-relevant to the query.

[REF1] - 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 fine tuned 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.

[REF2] - 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.

[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: 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.

[REF4] - 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.

[REF5] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Document_expansion_learning/BIBREF79_1225eb6570ce8d45067329fafcc8ff7636a65923.pdf
Title: Modeling and Solving Term Mismatch for Full-Text Retrieval Chunk of text: 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.

[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 out performs previous sparse models based on the comparable size of PLMs. The top-ranking performance of SparTerm even outper forms 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 representa tions 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 first stage 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/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.

[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: 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.

[REF9] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Document_expansion_learning/BIBREF79_1225eb6570ce8d45067329fafcc8ff7636a65923.pdf
Title: Modeling and Solving Term Mismatch for Full-Text Retrieval Chunk of text: Further speedup is possible using a Reverted Index (Pickens et al. 2010) that specializes in fast feedback retrieval. SVD and dependency parsing are the next most expensive tasks. Per query, SVD on 600 documents with 30,000 terms takes about 8 seconds (wall clock and CPU time). Further speedup is possible. Since the goal of SVD is just to find possible searchonyms of query terms, SVD

does not need to converge; a smaller number of iterations may suffice. Dependency parsing takes about 1 second per query (15 words) using the Stanford parser which not only does dependency parsing but also a full syntactic constituent parsing. Faster versions of dedicated dependency parsers exist, e.g. the MST Parser (McDonald et al. 2005).

.....

Title: Learned Sparse Retrieval - Impact score learning

In the field of information retrieval (IR), the retrieval performance is often measured using various metrics such as precision and recall [REF0]. One important aspect of retrieval is the scoring of documents based on their relevance to a given query. Traditionally, retrieval systems have used term-document similarity functions to compute the relevance scores [REF0]. However, these methods can be computationally expensive, especially when re-ranking a large number of documents [REF2]. To address this issue, researchers have proposed learned sparse retrieval techniques that aim to improve the efficiency of query processing while maintaining retrieval quality [REF2].

One approach to learned sparse retrieval is impact score learning, which focuses on learning the importance of individual terms in a document for retrieval [REF4]. The idea is to assign impact scores to terms based on their relevance to the query, and then use these scores to compute the overall relevance score of a document [REF4]. The impact scores can be learned using various techniques, such as deep neural networks [REF4]. By learning the impact scores, the retrieval system can prioritize the most important terms in a document, leading to more efficient query processing and improved retrieval performance [REF4].

To store the impact scores efficiently, quantization techniques can be applied [REF1]. Quantization involves representing the impact scores in a compressed form, which reduces the space requirements of the index [REF1]. For example, linear quantization with a fixed number of bits can be used to represent the impact scores within a specific range [REF1]. Experimental results have shown that quantization does not significantly affect the precision of the retrieval system [REF1]. At query processing time, the quantized impact scores of the document terms matching the query can be summed up to compute the query-document score [REF1].

The learned sparse retrieval techniques, including impact score learning, have been evaluated extensively in realistic settings using standard test collections and query logs [REF1]. These evaluations have compared the performance of the proposed methods against state-of-the-art baselines [REF1]. The experiments have shown that learned sparse retrieval techniques can achieve comparable or even better retrieval performance while reducing the computational cost of query processing [REF2]. However, there is still ongoing research to optimize the query processing speed of these techniques [REF7].

In conclusion, learned sparse retrieval techniques, particularly impact score learning, have shown promise in improving the efficiency of query processing in information retrieval systems. By learning the importance of individual terms in a document, these techniques can prioritize the most relevant terms and reduce the computational cost of query processing. Quantization techniques can be applied to store the impact scores efficiently. Experimental evaluations have demonstrated the effectiveness of these techniques in realistic settings. Further research is needed to optimize the query processing speed and explore new applications of learned sparse retrieval in information retrieval.

References given to GPT:

[REF0] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Impact_score_learning/BIBREF3_629f50daebbb9003f645f671f76cc6b33088c17d.pdfTitle: 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: Since storing a floating point value per posting would blow up the space requirements of the in verted 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 repro ducible setting, using state-of-the-art baselines and a standard test collection and query logs. Hardware.

[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: However, sev eral 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 com putational 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.

[REF3] - 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-to rank, a retrieval technique known to benefit the search engine's effectiveness by re-ranking a set of K documents.

[REF4] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Impact_score_learning/BIBREF91_4aa1d28944856ebe1950a27f633c6667ead3cbf8.pdf

Title: Learning Passage Impacts for Inverted Indexes Chunk of text: can perform processing in a single stage. Following a different paradigm, Dai and Callan investigated the use of the contextual word representations from BERT to gen erate more effective document term weights for bag-of-words re trieval. DeepCT , for passages, and HDCT , for documents, esti mate a term's context-specific importance in each passage/document, by projecting each word's BERT representation into a single term weight. These term weights are then transformed into term frequency like integer values that can be stored in an inverted index to be used with classical retrieval models. A main limitation of DeepCT that we address in this work is that it is trained as a per-token regression task, in which a ground truth term weight for every word is needed, and which does not permit the individual impact scores to co-adapt for the downstream objective of identifying relevant documents. By storing new integer values as term frequencies in the in verted

index, DeepCT and HDCT enrich a document's bag-of-words representation with additional document-level context informa tion, to match queries more accurately.

[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: 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, Deep Impact 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.

[REF6] - 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".

[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: We also see that DeepImpact mean response time exceeds the time reported for other methods. We trace this to the query pro cessing 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 per formed additional experiments using disjunctive query processing without optimizations, omitted for space limitations. These experi ments 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.

[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: the baseline, in terms of mean response time, and/or its (work) reduction, defined as the percentage of postings that are dynamically pruned, i.e., not scored, w.r.t. the baseline. When comparing two time quantities t1 and t2, with t1 > t2 we will always report their relative speedup s, defined as s = t1/t2 (always greater than 1). For example, if two strategies A and B have an average response time of 20 ms and 8 ms, respectively, their speedup (of B Full text available at: http://dx.doi.org/10.1561/150000005712 Introduction w.r.t. to A) is s = tA/tB = 20/8 = 2.5×. When comparing two numbers of processed elements n1 and n2, with n1 > n2 we will systematically report the percentage reduction r, defined as r = 1−n2/n1.

[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 various fields, including information retrieval and machine learning. It involves finding a compact and informative representation of data by encouraging sparsity in the learned representations. In this section, we discuss the concept of learned sparse retrieval and its application in sparse representation learning.

One approach to learned sparse retrieval is to optimize the expected FLOPs (Floating Point Operations) while minimizing the loss function [REF0]. Since the distribution of data is often unknown, empirical estimates are used to approximate the FLOPs. For example, the empirical fraction of non-zero activations can be used as a consistent estimator for FLOPs [REF0]. By controlling the expected FLOPs, the goal is to achieve a balance between model efficiency and effectiveness.

Another approach to learned sparse retrieval is through the use of regularization techniques. Regularization helps in controlling the sparsity of the learned representations and improving the efficiency and effectiveness of the models [REF1]. For example, SparTerm model predicts term importance based on the logits of the Masked Language Model (MLM) layer and applies regularization to ensure the positivity of term weights [REF1]. By incorporating regularization into the learning process, the models can achieve better sparsity and performance.

Sparse lexical representations have also been explored in the context of learned sparse retrieval. These representations aim to capture the importance of individual terms in a query or document sequence [REF1]. For instance, the SparTerm model predicts term importance based on BERT embeddings and applies ReLU activation to ensure the positivity of term weights [REF1]. By considering the importance of individual terms, sparse lexical representations can provide valuable insights into the underlying structure of the data.

Efficient approximate nearest-neighbor search is another area where learned sparse retrieval has been applied. In high-dimensional spaces, exact retrieval of nearest neighbors can be computationally expensive. Therefore, approximate methods are often used to trade off accuracy for efficiency [REF5]. Approaches such as Locality Sensitive Hashing (LSH), Navigable Small World Graphs (NSW), and Product Quantization (PQ) have been employed to learn compact and sparse representations that preserve distance information [REF9]. These methods leverage carefully chosen data structures and dimensionality reduction techniques to speed up query times while maintaining the effectiveness of retrieval.

In summary, learned sparse retrieval and sparse representation learning have emerged as powerful techniques in various domains. By optimizing the expected FLOPs, incorporating regularization, and leveraging sparse lexical representations, these approaches enable efficient and effective retrieval while maintaining the interpretability of the learned representations. Additionally, in the context of approximate nearest-neighbor search, learned sparse retrieval techniques provide a trade-off between accuracy and efficiency, making them suitable for large-scale applications.

References given to GPT:
[REF0] - 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) +

[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 vocab ulary (|V | = 30522) – based on the logits of the Masked Lan guage 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 acti vation 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 im portance 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/BIBREF95_1e8a6de5561f557ff9abf43d538d8d5e9347efa0.pdf
Title: SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking Chunk of text: 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. Results are given in Table 1. Overall, we observe that: (1) our mod els outperform the other sparse retrieval methods by a large margin (except for recall@1000 on TREC DL); (2) the results are competitive with state-of-the-art dense retrieval methods. More specifically, our training method for ST lexical-only already outperforms the results of DeepCT as well as the results reported in the original SparTerm paper – including the model using expansion. Thanks to the additional sparse expansion mechanism, we are able to obtain results on par with state-of-the-art dense approaches on MS MARCO dev set (e.g. Recall@1000 close to 0.96 for ST exp-ℓ1), but with a much bigger average number of FLOPS. By adding a log-saturation effect to the expansion model, SPLADE greatly increases sparsity – reducing the FLOPS to similar levels 2We made the code public at https://github.com/naver/splade Short Research Paper III SIGIR '21, July 11–15, 2021, Virtual Event, Canada 2290Table 1: Evaluation on MS MARCO passage retrieval (dev set) and TREC DL 2019 model MS MARCO dev TREC DL 2019 FLOPS MRR@10 R@1000 NDCG@10 R@1000 Dense retrieval Siamese (ours) 0.312 0.941 0.637 0.711

[REF3] - 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 propose a new approach for passage retrieval that performs modeling of term importance (i.e., salience) and expansion over a contextualized language model to build query and document repre sentations. We call this approach EPIC (Expansion via Prediction of Importance with Contextualization). At query time, EPIC can be employed as an inexpensive re-ranking method because document representations can be pre-computed at index time. EPIC improves upon the prior state of the art on the MS-MARCO passage ranking dataset by substantially narrowing the effectiveness gap between practical approaches with subsecond retrieval times and those that are considerably more expensive, e.g., those using BERT as a re ranker. Furthermore, the proposed representations are interpretable because the dimensions of the representation directly correspond to the terms in the lexicon. An overview is shown in Fig. 1.

[REF4] - paperID: ./papers_pdf/paper_section/Learned_Sparse_Retrieval-Sparse_representation_learning/BIBREF95_1e8a6de5561f557ff9abf43d538d8d5e9347efa0.pdf
Title: SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking Chunk of text: This gives the following regularization loss ℓFLOPS = Õ j ∈V a¯ 2 j = Õ j ∈V 1 N Õ N i=1 w (di) j !2 This differs from the ℓ1 regularization used in SNRM where the a¯j are not squared: using

ℓFLOPS thus pushes down high average term weight values, giving rise to a more balanced index. Overall loss. We propose to combine the best of both worlds for end-to-end training of sparse, expansion-aware representations of documents and queries. Thus, we discard the binary gating in SparTerm, and instead learn our log-saturated model (Eq. 4) by jointly optimizing ranking and regularization losses: L = L$rank-IBN + \lambda q$L

[REF5] - 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: Since accurate search in high dimensions is prohibitively expensive in practice (Wang, 2011), one has to typically sacrifice accuracy for efficiency by resorting to approximate methods. Addressing the problem of efficient approximate Nearest-Neighbor Search (NNS) (Jegou et al., 2011) or Maximum Inner-Product Search (MIPS) (Shrivastava and Li, 2014) is thus an active area of research, which we review in brief in the related work section. Most approaches (Charikar, 2002; Jegou et al., 2011) aim to learn compact lower-dimensional representations that preserve distance information. While there has been ample work on learning compact representations, learning sparse higher dimensional representations have been addressed only recently (Jeong and Song, 2018; Cao et al., 2018). As a seminal instance, Jeong and Song (2018) propose an end-to-end approach to learn ∗ Part of the work was done when BP was a research intern at Snap Inc. 1The implementation is available at https://github.com/biswajitsc/sparse-embed 1 arXiv:2004.05665v1 [cs.LG] 12 Apr 2020Published as a conference paper at ICLR 2020 sparse and high-dimensional hashes, showing significant speed-up in retrieval time on benchmark datasets compared to dense embeddings.

[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: 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 com putation 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.

[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: Define the mean absolute activation aj = E[|fθ(X)j |] and its empirical version a¯j = 1 n Pn i=1 |fθ(xi)j |, which is the `1 norm of the activations (scaled by 1/n) in contrast to the `0 quasi norm in the FLOPs calculation. Define the relaxations, Fe(fθ,P) = Pd j=1 a 2 j and its consistent estimator Fe(fθ, D) = Pd j=1 a¯ 2 j . We propose to minimize the following relaxation, which can be optimized using any off-the-shelf stochastic gradient 5Published as a conference paper at ICLR 2020 descent optimizer. min θ∈Θ `(fθ, D) + λFe(fθ, D) | {z } Le(θ) . (4) Sparse retrieval and re-ranking.

[REF8] - 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: The elements of ϕq that correspond to to terms not in the query are set to 0. For each term ti appearing in the t1, . . . ,tn terms of the query q, the corresponding element ϕq(ti) is equal to the importance wq(ti) of the term w.r.t. the query wq(ti) = ln 1 + softplus θ ⊤ 1 fi(q) , (1) where θ1 ∈ R e is a vector of learned parameters. The softplus(·) function is defined as softplus(x) = ln(1 + e x). The use of softplus ensures that no terms have a negative importance score, while im posing no upper bound.

[REF9] - 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.

.....