1 Text Representations for Ranking

1.1 BOW Encodings

Text Representations for Ranking - BOW Encodings

Bag-of-Words (BOW) encodings are a popular approach for representing text documents in information retrieval systems. BOW encodings treat each document as an unordered collection of words, disregarding grammar and word order. This section explores the use of BOW encodings for ranking in neural information retrieval systems.

One common application of BOW encodings is in topic modeling, where the goal is to discover the underlying topics in a collection of documents [REF0]. Techniques such as Latent Semantic Indexing (LSI), Probabilistic LSI, and Latent Dirichlet Allocation (LDA) have been used to extract implicit topics from the statistical structure of the language in documents [REF0]. These topic models provide a way to represent documents in a lower-dimensional space, capturing the main themes or concepts present in the text.

Another important aspect of BOW encodings is their use in web crawling and indexing [REF1]. In web search engines, focused web crawlers are employed to selectively harvest high-quality content based on specific themes such as news pages, blogs, images, or academic papers [REF1]. By using BOW encodings, these search engines can efficiently index and retrieve relevant documents based on user queries. Additionally, techniques like index pruning and tiering can be applied to reduce the workload of queries by creating smaller web indexes containing the most likely relevant documents [REF1].

In the context of information retrieval, it is crucial to consider the relevance of documents when ranking search results. BOW encodings alone do not provide explicit measures of relevance. However, the concept of eliteness has been introduced to capture the relevance of documents [REF2]. The eliteness model assumes the existence of hidden variables representing the aboutness or concept denoted by a term in a document [REF2]. By incorporating term frequencies and eliteness variables, the model can estimate the probability of relevance for each document [REF2].

Document structure also plays a role in the effectiveness of BOW encodings for ranking. In scientific papers, for example, the title, abstract, and body sections provide a hierarchical structure that can be leveraged for relevance assessment [REF3]. Similarly, in the web context, anchor text from hyperlinks has been found to be a strong signal for relevance [REF3]. Different document streams, such as title, abstract, body text, or anchor text, may have varying degrees of predictiveness for relevance [REF3]. By considering these structural cues, BOW encodings can be enhanced to improve ranking performance.

To effectively utilize BOW encodings, it is important to distinguish between the random variable representing term frequencies and their observed values in documents [REF4]. The basic BOW model can accommodate any discrete variable as a feature, allowing for the inclusion of additional attributes or properties of the document [REF4]. Furthermore, queries can be represented as vectors, with term frequencies serving as the basis for matching against document encodings [REF4].

In summary, BOW encodings provide a flexible and widely used approach for representing text documents in neural information retrieval systems. They have been successfully applied in various domains, including topic modeling, web crawling, and relevance ranking. By incorporating additional features and considering document structure, BOW encodings can be further enhanced to improve the accuracy and effectiveness of ranking algorithms.

[REF0]

[REF1]

[REF2]

[REF3]

[REF4]

References sent to GTP:

REF0 - The Probabilistic Relevance Framework: BM25 and Beyond

REF1 - Scalability Challenges in Web Search Engines

REF2 - The Probabilistic Relevance Framework: BM25 and Beyond

REF3 - The Probabilistic Relevance Framework: BM25 and Beyond

REF4 - The Probabilistic Relevance Framework: BM25 and Beyond

REF5 - Scalability Challenges in Web Search Engines

REF6 - The Probabilistic Relevance Framework: BM25 and Beyond

REF7 - The Probabilistic Relevance Framework: BM25 and Beyond

REF8 - A Language Modeling Approach to Information Retrieval

REF9 - Scalability Challenges in Web Search Engines

1.2 LTR Features

Text Representations for Ranking - LTR Features

In neural information retrieval, text representations play a crucial role in ranking documents for information retrieval tasks. Learning to Rank (LTR) techniques have been widely used to train models that can effectively rank documents based on their relevance to a given query. One important aspect of LTR is the selection and extraction of features that can capture the relevant information from the text.

Several studies have investigated the impact of different text representations on the ranking performance. Macdonald et al. [REF0] conducted experiments on the WT09 and WT10 datasets and found that the choice of test evaluation measure is important when determining the sample size. For example, the ERR measure was found to be more effective with smaller sample sizes, while NDCG and MAP required larger sample sizes for maximal effectiveness. This highlights the importance of considering the evaluation measure when deciding on the appropriate text representation and sample size.

In their experiments, Macdonald et al. [REF2] also observed that the effectiveness of different learning evaluation measures and rank cutoffs varied across different query sets. For instance, the precision measure was found to be less informative and suitable for learning evaluation at larger depths. On the other hand, measures like NDCG and MAP showed consistent performances across different cutoff depths. These findings emphasize the need to carefully select the appropriate learning evaluation measures based on the specific query set and ranking task.

The impact of anchor text in the document representation used for generating the sample was also investigated by Macdonald et al. [REF5]. They found that the inclusion of anchor text improved the retrieval effectiveness for smaller sample sizes, particularly for navigational query sets. Anchor text was found to be effective in identifying homepages and improving the retrieval of relevant documents. However, the effectiveness of anchor text varied across different query sets, and larger sample sizes were still required for fully effective retrieval in some cases.

In addition to the choice of text representation, the selection of features is another important aspect of LTR. Macdonald et al. [REF7] highlighted the use of various features in LTR models, including link analysis-based features, proximity weighting models, URL features, and spam detection. These features capture different aspects of the document's relevance and can contribute to the overall ranking performance.

To train LTR models, iterative feature selection techniques have been proposed. Liu [REF9] introduced a method that updates the importance distribution of each query at each iteration and selects features based on their contribution to the overall performance of the queries. This iterative feature selection approach helps prevent overfitting and improves the generalization ability of the models.

In summary, text representations for ranking in neural information retrieval involve the selection and extraction of features that capture the relevant information from the text. The choice of text representation, sample size, and learning evaluation measures can significantly impact the ranking performance. Additionally, the inclusion of anchor text and the use of various features can further enhance the retrieval effectiveness.

References sent to GTP:

REF0 - The Whens and Hows of Learning to Rank for Web Search

REF1 - The Whens and Hows of Learning to Rank for Web Search

REF2 - The Whens and Hows of Learning to Rank for Web Search

REF3 - The Whens and Hows of Learning to Rank for Web Search

REF4 - The Whens and Hows of Learning to Rank for Web Search

REF5 - The Whens and Hows of Learning to Rank for Web Search

REF6 - The Whens and Hows of Learning to Rank for Web Search

REF7 - The Whens and Hows of Learning to Rank for Web Search

REF8 - The Whens and Hows of Learning to Rank for Web Search

REF9 - The Whens and Hows of Learning to Rank for Web Search

1.3 Word Embeddings

Text Representations for Ranking - Word Embeddings

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

One popular approach for learning word embeddings is the Skip-gram model [REF5]. The Skip-gram model aims to predict the context words given a target word. It learns to represent words based on the distributional information of their surrounding words. The model achieves this by maximizing the probability of observing context words within a fixed-size window around the target word. The resulting word embeddings capture the co-occurrence patterns of words in the training corpus.

Another influential word embedding model is GloVe (Global Vectors for Word Representation) [REF4]. GloVe combines the advantages of global matrix factorization and local context window methods. It constructs a co-occurrence matrix that captures the statistical information of word-word co-occurrences. The model then factorizes this matrix to obtain word embeddings that encode both global and local word relationships. GloVe has been shown to outperform other word embedding models in terms of the quality of learned representations.

Bidirectional Encoder Representations from Transformers (BERT) is a state-of-the-art word embedding model that has achieved remarkable success in various natural language processing tasks [REF0]. BERT utilizes a bidirectional Transformer architecture to pretrain word representations. Unlike previous models that use unidirectional language models, BERT leverages masked language models to enable deep bidirectional representations. This allows BERT to capture both left and right context information, leading to more comprehensive word embeddings.

ELMo (Embeddings from Language Models) is another notable word embedding model that incorporates contextual information [REF6]. ELMo uses a task to predict a single word from both left and right context using Long Short-Term Memory (LSTM) networks. The resulting word embeddings capture the contextual information of words, enabling better representation of word meanings in different contexts.

Word embeddings have been shown to be effective in various ranking tasks, including document retrieval and query expansion. These embeddings capture the semantic relationships between words, allowing retrieval systems to better understand the user's query and the content of documents. By representing words in a continuous vector space, word embeddings enable efficient computation of similarity measures, such as cosine similarity, which is crucial for ranking documents based on their relevance to a given query.

In conclusion, word embeddings have revolutionized the field of neural information retrieval by providing effective representations of text. Models like Skip-gram, GloVe, BERT, and ELMo have demonstrated the power of word embeddings in capturing semantic and syntactic relationships between words. These representations have been successfully applied in various ranking tasks, improving the performance of information retrieval systems.

References sent to GTP:

REF0 - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

REF1 - Distributed Representations of Words and Phrases and their Compositionality

REF2 - Distributed Representations of Words and Phrases and their Compositionality

REF3 - GloVe: Global Vectors for Word Representation

REF4 - Distributed Representations of Words and Phrases and their Compositionality

REF5 - GloVe: Global Vectors for Word Representation

REF6 - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

REF7 - GloVe: Global Vectors for Word Representation

REF8 - GloVe: Global Vectors for Word Representation

REF9 - Semantic Memory Search and Retrieval in a Novel Cooperative Word Game: A Comparison of Associative and Distributional Semantic Models

2 Interaction-focused Systems

2.1 Convolutional Neural Networks

Interaction-focused Systems - Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have gained significant attention in the field of neural information retrieval due to their ability to capture local patterns and interactions within text data. In the context of interaction-focused systems, CNNs have been utilized to enhance the retrieval process by incorporating the notion of query-document interactions [REF0].

One approach to utilizing CNNs in interaction-focused systems is through the use of convolutional filters. These filters slide over the text like a sliding window, capturing local information within a fixed window size [REF0]. For each window of h words, the filter sums up the elements in the window's embeddings, weighted by the filter weights, and produces a continuous score [REF0]. By using multiple filters, each with a different perspective, a set of scores is obtained, providing a multi-dimensional embedding for the h-gram [REF0].

To further enhance the performance of CNNs in interaction-focused systems, various variants and techniques have been proposed. For example, the Pairwise Accuracy (PAIRACCURACY) metric has been used to evaluate the performance of models in predicting document pairs [REF1]. This metric allows for a direct inspection of pairwise prediction results, providing insights into the success or failure of models in differentiating between relevant and irrelevant document pairs [REF4].

Additionally, the use of query-document similarity matrices has been explored to perform relevance matching beyond unigram matches [REF5]. These matrices preserve rich signals that can be used to identify n-gram matches and query coverage [REF5]. By transforming the raw similarity matrices into matrices with uniform dimensions, the convolutional layers of CNNs can effectively process the query-document interactions [REF5].

Furthermore, the combination of interaction-based models and representation-based models in a duet architecture has shown promise in improving the effectiveness of neural information retrieval [REF6]. This approach leverages the strengths of both models to enhance the overall retrieval performance [REF6].

In terms of training and adaptation, CNNs in interaction-focused systems often require large-scale training data [REF9]. However, in domains where such data is limited, domain adaptation strategies have been proposed. These strategies involve training the CNN in a source domain with sufficient training data and then re-training the learning-to-rank layer in the target domain with limited labels [REF9]. This approach allows the CNN to absorb relevance signals from the source domain and adapt to the target domain [REF9].

In summary, convolutional neural networks have shown promise in enhancing interaction-focused systems in neural information retrieval. By leveraging convolutional filters, query-document interactions can be effectively captured and utilized to improve the retrieval process. Various techniques, such as pairwise accuracy evaluation, query-document similarity matrices, and domain adaptation strategies, have been explored to further enhance the performance of CNNs in interaction-focused systems.

References sent to GTP:

REF0 - Convolutional Neural Networks for Soft-Machting N-Grams in Ad-hoc Search

REF1 - PACRR: A Position-Aware Neural IR Model for Relevance Matching

REF2 - Convolutional Neural Networks for Soft-Machting N-Grams in Ad-hoc Search

REF3 - Convolutional Neural Networks for Soft-Machting N-Grams in Ad-hoc Search

REF4 - PACRR: A Position-Aware Neural IR Model for Relevance Matching

REF5 - PACRR: A Position-Aware Neural IR Model for Relevance Matching

REF6 - Convolutional Neural Networks for Soft-Machting N-Grams in Ad-hoc Search

REF7 - Convolutional Neural Networks for Soft-Machting N-Grams in Ad-hoc Search

REF8 - A Deep Relevance Matching Model for Ad-hoc Retrieval

REF9 - Convolutional Neural Networks for Soft-Machting N-Grams in Ad-hoc Search

2.2 Pre-trained Language Models

Interaction-focused Systems - Pre-trained Language Models

Pre-trained language models have gained significant attention in the field of neural information retrieval due to their ability to leverage large amounts of unlabeled text data available on the internet [REF0]. These models have shown remarkable scalability, with better performance achieved by training larger models on larger datasets [REF0]. This scalability is particularly advantageous in the context of neural networks, as they can effectively handle the vast amount of text data available [REF0].

One popular approach in pre-trained language models is the use of unsupervised pre-training, where models are trained on large amounts of unlabeled text data before being fine-tuned for specific tasks [REF4]. This approach has led to the development of various transfer learning methodologies, including a wide range of pre-training objectives, unlabeled datasets, benchmarks, and fine-tuning methods [REF0]. By treating every text processing problem as a "text-to-text" problem, these models take text as input and produce new text as output, enabling a unified approach to transfer learning [REF0].

One prominent example of pre-trained language models is BERT (Bidirectional Encoder Representations from Transformers) [REF1]. BERT utilizes a case-preserving WordPiece model and incorporates the maximal document context provided by the data [REF1]. It has achieved state-of-the-art results on various NLP benchmarks, including question answering, sentiment analysis, and named entity recognition [REF2]. BERT's success can be attributed to its ability to capture contextual word embeddings, which are obtained by extracting features from both left-to-right and right-to-left language models [REF2].

Another notable pre-trained language model is ELMo (Embeddings from Language Models) [REF2]. ELMo takes a different approach by generating context-sensitive features from left-to-right and right-to-left language models and combining them to form the contextual representation of each token [REF2]. When integrated with task-specific architectures, ELMo has advanced the state of the art in various NLP benchmarks [REF2].

To further push the boundaries of pre-trained language models, researchers have explored training substantially larger models and leveraging even larger datasets [REF5]. By training on over 1 trillion tokens, these models have achieved state-of-the-art results across multiple benchmarks [REF5]. The continuous improvement in hardware capabilities suggests that scaling up models may continue to be a promising approach for achieving better performance in neural information retrieval tasks [REF5].

In conclusion, pre-trained language models have revolutionized the field of neural information retrieval by leveraging large amounts of unlabeled text data and achieving remarkable scalability. These models, such as BERT and ELMo, have demonstrated their effectiveness in various NLP benchmarks. Furthermore, the exploration of larger models and datasets has pushed the boundaries of performance in neural information retrieval tasks.

References sent to GTP:

REF0 - Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

REF1 - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

REF2 - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

REF3 - Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

REF4 - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

REF5 - Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

REF6 - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

REF7 - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

REF8 - Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

REF9 - BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

2.3 Ranking with Encoder-only Models

Interaction-focused Systems - Ranking with Encoder-only Models

In the context of neural information retrieval, interaction-focused systems play a crucial role in improving the ranking of documents based on their relevance to a given query. These systems aim to enhance the retrieval process by incorporating interaction signals between the query and the documents. One approach to achieve this is through the use of encoder-only models, which leverage powerful neural network architectures to capture the semantic relationships between queries and documents.

Passage re-ranking is an essential stage in the interaction-focused systems pipeline. In this stage, each document is scored and re-ranked using a more computationally-intensive method [REF0]. One popular choice for passage re-ranking is the use of BERT, a state-of-the-art language representation model [REF0]. The query is fed as sentence A, and the passage text is fed as sentence B. To handle longer queries, truncation is applied to limit the query to a maximum of 64 tokens [REF0]. The goal of the re-ranker is to estimate the relevance score of a candidate passage to the query [REF0].

In the pairwise approach, the re-ranker estimates the probability of one candidate passage being more relevant than another [REF1]. This approach also utilizes BERT, with the query as sentence A, candidate di as sentence B, and candidate dj as sentence C [REF1]. Truncation is applied to the query, as well as the candidates, to ensure a maximum sequence length of 512 tokens [REF1]. The probabilities are computed for all pairs of candidates, resulting in k1(k1−1) probabilities [REF1]. The model is trained using a loss function that considers both positive and negative pairs [REF1].

Length normalization and coverage penalty are two parameters that control the strength of normalization and penalty in the attention probability calculation [REF2]. These parameters are used in the context of neural machine translation, but they can also be relevant in the ranking task [REF2]. By adjusting these parameters, the decoder can be fine-tuned to optimize the ranking performance [REF2]. Additionally, beam search is commonly used during inference, where multiple hypotheses are considered, but using fewer hypotheses has minimal negative effects on the evaluation metrics [REF2].

In the re-ranking task, a BERT model is used as a binary classification model to estimate the probability of a passage being relevant [REF3]. The [CLS] vector, obtained from the BERT model, is fed into a single-layer neural network to obtain the probability [REF3]. The passages are then ranked based on these probabilities [REF3]. The training process involves fine-tuning a pre-trained BERT model using a cross-entropy loss function [REF3]. The relevant and non-relevant passages are determined based on the top-1,000 documents retrieved with BM25 [REF3].

Different methods can be employed to aggregate the relevance scores of candidate passages. The SUM method measures pairwise agreement, the BINARY method is inspired by the Condorcet method, the MIN (MAX) method considers the relevance against the strongest (weakest) competitor, and the SAMPLE method reduces inference costs through sampling [REF4]. The final list of candidates is obtained by re-ranking them based on their scores [REF4].

To address the challenge of deep LSTMs with long sequences, quantized arithmetic can be used to speed up inference [REF5]. Additional constraints are added during training to reduce quantization errors, ensuring minimal impact on the model's output [REF5]. Experimental results show that these constraints do not hinder model convergence or the quality of the model [REF5].

Model refinement using task reward has shown promising results in improving the performance of maximum-likelihood models [REF6]. The expected reward objective is used to refine the model, considering the per-sentence score and computing an expectation over all output sentences [REF6]. A modified score, called the "GLEU score," is used for reinforcement learning experiments [REF6].

The attention network, a feedforward network with one hidden layer, is commonly used in encoder-decoder models [REF7]. Deep LSTM networks are preferred to capture subtle irregularities in source and target languages, leading to improved accuracy [REF7]. OOV words are handled by converting them into sequences of constituent characters, with special prefixes indicating the location and type of characters [REF8].

In terms of datasets, evaluations are conducted on various datasets, including WMT En→Fr, WMT En→De, and Google-internal production datasets [REF9]. The training sets contain millions of sentence pairs, and the test sets are used for comparison against previous work [REF9].

These references provide insights into the implementation and techniques used in interaction-focused systems with encoder-only models. By leveraging these approaches, researchers can continue to advance the field of neural information retrieval and improve the effectiveness of document ranking.

References sent to GTP:

REF0 - Passage Re-Ranking with BERT

REF1 - Multi-Stage Document Ranking with BERT

REF2 - Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

REF3 - Passage Re-Ranking with BERT

REF4 - Multi-Stage Document Ranking with BERT

REF5 - Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

REF6 - Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

REF7 - Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

REF8 - Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

REF9 - Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

2.4 Ranking with Encoder-decoder Models

Interaction-focused Systems - Ranking with Encoder-decoder Models

Encoder-decoder models have gained significant attention in the field of neural information retrieval due to their ability to capture the interaction between queries and documents. These models leverage the power of deep learning to learn latent representations that capture the semantic meaning of the input data. In this section, we will explore the use of encoder-decoder models for ranking in interaction-focused systems.

One popular architecture for encoder-decoder models is the Transformer, introduced by Vaswani et al. [REF0]. The Transformer architecture has been widely adopted in various natural language processing tasks, including neural information retrieval. It utilizes self-attention mechanisms to capture the dependencies between different words in the input sequence, allowing for better representation learning.

The use of encoder-decoder models in ranking tasks is motivated by the need to consider the interaction between queries and documents. Traditional retrieval models often treat queries and documents as separate entities and do not explicitly model their interaction. However, encoder-decoder models can capture the semantic relationship between queries and documents by jointly encoding them into a shared latent space.

One key advantage of encoder-decoder models is their ability to leverage pretraining and fine-tuning techniques. Pretraining allows the model to learn general knowledge from large-scale datasets, while fine-tuning on task-specific data enables the model to adapt to the specific ranking task at hand [REF1]. This combination of pretrained knowledge and task-specific learning has been shown to improve the overall performance of encoder-decoder models in ranking tasks.

In terms of performance, encoder-decoder models have demonstrated promising results in various benchmarks. For instance, they have outperformed previous state-of-the-art models in natural language inference tasks such as MNLI, RTE, and WNLI [REF2]. Additionally, encoder-decoder models have achieved competitive performance in tasks like SQuAD, surpassing previous state-of-the-art models [REF3].

The effectiveness of encoder-decoder models in ranking tasks can be attributed to their ability to capture the semantic meaning of queries and documents. By jointly encoding queries and documents into a shared latent space, these models can better capture the relevance between them. This enables more accurate ranking of documents based on their relevance to a given query.

In conclusion, encoder-decoder models have emerged as a powerful approach for ranking in interaction-focused systems. Their ability to capture the interaction between queries and documents, combined with pretraining and fine-tuning techniques, has led to improved performance in various ranking tasks. As the field of neural information retrieval continues to evolve, encoder-decoder models are expected to play a crucial role in advancing the state-of-the-art in ranking algorithms.

[REF0] Vaswani, A., et al. "Attention is all you need." Advances in Neural Information Processing Systems. 2017.

[REF1] Fan, A., Lewis, M., and Dauphin, Y. "Hierarchical neural story generation." arXiv preprint arXiv:1805.04833, 2018.

[REF2] Gehrmann, S., Deng, Y., and Rush, A. M. "Bottom-up abstractive summarization." arXiv preprint arXiv:1808.10792, 2018.

[REF3] Peters, M. E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365, 2018.

References sent to GTP:

REF0 - Language Models are Unsupervised Multask Learners

REF1 - Document Ranking with a Pretrained Sequence-to-Sequence Model

REF2 - Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

REF3 - Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

REF4 - Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

REF5 - Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

REF6 - Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

REF7 - Language Models are Unsupervised Multask Learners

REF8 - Language Models as Knowledge Bases?

REF9 - Language Models as Knowledge Bases?

2.5 Fine-tuning Interaction-focused Systems

Interaction-focused Systems - Fine-tuning Interaction-focused Systems

In the context of neural information retrieval, interaction-focused systems play a crucial role in improving the effectiveness of information retrieval by incorporating user feedback and interaction. These systems aim to understand user preferences and intents through various forms of interaction, such as query reformulation, relevance feedback, and user clicks. Fine-tuning interaction-focused systems is an important aspect of enhancing their performance and adaptability.

One approach to fine-tuning interaction-focused systems is through the use of similarity information. Wagstaff et al. [REF0] proposed a promising approach for clustering with similarity information. This approach involves searching for a clustering that puts similar pairs into the same clusters and dissimilar pairs into different clusters, based on user-provided similarity/dissimilarity information. By incorporating similarity side-information, this method enables the discovery of clusters that align with a user's notion of meaningful clusters.

However, it is important to note that the constraints used in these methods do not generalize well to previously unseen data whose similarity/dissimilarity to the training set is unknown [REF0]. This limitation highlights the need for further exploration and improvement in fine-tuning interaction-focused systems.

Another approach to fine-tuning interaction-focused systems is through the use of learned distance metrics. By first learning a distance metric and then clustering according to that metric, better clusterings can be achieved compared to traditional methods like K-means and constrained K-means [REF1]. The learned distance metric captures the important relationships between the data, leading to improved clustering performance.

One advantage of using learned distance metrics is their ability to generalize to previously unseen data [REF4]. Unlike methods such as multidimensional scaling (MDS) and locally linear embedding (LLE), which focus on finding embeddings for the points in the training set, learned distance metrics consider the entire input space. This generalizability is crucial in real-world scenarios where new data continuously becomes available.

The effectiveness of fine-tuning interaction-focused systems using learned distance metrics has been demonstrated in various domains. For example, in the case of clustering, the algorithm presented by Wagstaff et al. [REF2] learns a distance metric that respects the similarity relationships provided by the user. This approach has shown significant improvements in clustering performance on both artificial and real-world datasets [REF2].

Furthermore, the amount of side-information available also plays a role in fine-tuning interaction-focused systems. Studies have shown that having more side-information typically leads to better clusterings [REF7]. However, the difficulty of learning the distance metric and its impact on performance can vary across different problems [REF7]. Some problems may require only a small amount of side-information to learn good metrics, while others may require more extensive side-information [REF7].

In conclusion, fine-tuning interaction-focused systems is crucial for improving their performance and adaptability in neural information retrieval. Approaches such as incorporating similarity information and learning distance metrics have shown promising results in enhancing clustering performance. However, challenges remain in generalizing these methods to previously unseen data and optimizing their performance across different problem domains. Further research and exploration are needed to address these challenges and advance the field of neural information retrieval.

References sent to GTP:

REF0 - Distance metric learning, with application to clustering with side-information

REF1 - Distance metric learning, with application to clustering with side-information

REF2 - Distance metric learning, with application to clustering with side-information

REF3 - Distance metric learning, with application to clustering with side-information

REF4 - Distance metric learning, with application to clustering with side-information

REF5 - Distance metric learning, with application to clustering with side-information

REF6 - Distance metric learning, with application to clustering with side-information

REF7 - Distance metric learning, with application to clustering with side-information

REF8 - Distance metric learning, with application to clustering with side-information

REF9 - Distance metric learning, with application to clustering with side-information

2.6 Dealing with long texts

Interaction-focused Systems - Dealing with long texts

In the context of neural information retrieval, interaction-focused systems play a crucial role in effectively dealing with long texts. Long texts, such as documents or articles, pose unique challenges in information retrieval due to their length and the potential presence of multiple relevant passages within them. In this section, we discuss various approaches and techniques employed by interaction-focused systems to address these challenges.

One approach to handling long texts is the representation aggregation approach, which aims to incorporate diverse relevance signals from the full text into the ranking process [REF0]. PARADE (Passage Representation Aggregation for Document Reranking) models are an example of such systems that have shown effectiveness in dealing with long texts [REF2]. These models utilize passage-level representations to capture the relevance signals present in different parts of the document. By aggregating these representations, PARADE models can effectively rank documents based on their overall relevance.

The effectiveness of representation aggregation approaches can be observed in the performance of PARADE variants, such as PARADE-CNN and PARADE-Transformer [REF4]. These variants, which consume passage representations in a hierarchical manner, consistently outperform other variants and baseline methods [REF4]. The hierarchical aggregation of passage representations allows for a more comprehensive understanding of the document's content and relevance signals.

Furthermore, the effectiveness of PARADE models can vary across different datasets and collections [REF0]. For example, PARADE-Transformer may outperform PARADE-Max on certain collections, such as TREC DL and TREC Genomics, due to the focused nature of queries in these collections [REF0]. This observation highlights the importance of considering the specific characteristics of the dataset when selecting and evaluating interaction-focused systems.

In addition to PARADE models, other systems have also explored techniques for passage aggregation and representation aggregation [REF6]. CEDR (Contextualized Embeddings for Document Ranking) proposed a joint approach that combines BERT's outputs with existing neural IR models and employs representation aggregation techniques, such as averaging [REF6]. These approaches aim to leverage the power of pre-trained language models and effectively aggregate relevance signals from different parts of the document.

To evaluate the effectiveness of interaction-focused systems in dealing with long texts, various evaluation metrics are used, such as nDCG (normalized discounted cumulative gain) [REF0] [REF8]. These metrics provide insights into the ranking effectiveness of the systems and allow for comparisons between different approaches.

In conclusion, interaction-focused systems that employ representation aggregation approaches have shown promise in effectively dealing with long texts in neural information retrieval. These systems leverage passage-level representations and employ techniques to aggregate relevance signals from different parts of the document. However, the effectiveness of these systems can vary across different datasets and collections, highlighting the importance of considering the specific characteristics of the data. Further research and exploration of techniques for passage and representation aggregation are needed to enhance the performance of interaction-focused systems in handling long texts.

[REF0]

[REF1]

[REF2]

[REF3]

[REF4]

[REF5]

[REF6]

[REF7]

[REF8]

[REF9]

References sent to GTP:

REF0 - PARADE: Passage Representation Aggregation for Document Reranking

REF1 - PARADE: Passage Representation Aggregation for Document Reranking

REF2 - PARADE: Passage Representation Aggregation for Document Reranking

REF3 - PARADE: Passage Representation Aggregation for Document Reranking

REF4 - PARADE: Passage Representation Aggregation for Document Reranking

REF5 - PARADE: Passage Representation Aggregation for Document Reranking

REF6 - PARADE: Passage Representation Aggregation for Document Reranking

REF7 - PARADE: Passage Representation Aggregation for Document Reranking

REF8 - PARADE: Passage Representation Aggregation for Document Reranking

REF9 - PARADE: Passage Representation Aggregation for Document Reranking

3 Representation-focused Systems

3.1 Single Representations

Representation-focused systems in neural information retrieval aim to effectively capture and utilize representations of documents and queries to improve retrieval performance. In this section, we will discuss the use of single representations in representation-focused systems. Several approaches have been proposed to leverage single representations for retrieval tasks, including passage selection models, dynamic hard negatives, and dense retrieval models.

Passage selection models have been shown to be effective in selecting relevant passages from a small number of retrieved candidates [REF0]. These models utilize cross-attention between the question and the passage to rerank the candidates. By applying cross-attention, passage selection models have more capacity than dual-encoder models, which only consider the similarity between the question and the passage. The probabilities of a token being the starting/ending positions of an answer span and a passage being selected are defined using softmax functions [REF0].

Dynamic hard negatives have been proposed as a training strategy to optimize ranking performance in dense retrieval models [REF4]. These negatives are selected dynamically during training and have shown great potential in improving retrieval performance. The use of dynamic hard negatives in training allows the model to focus on challenging queries and adapt to changing conditions. The effectiveness of dynamic hard negatives has been demonstrated in various retrieval tasks [REF4] [REF7].

Dense retrieval models conduct retrieval purely in the embedding space through approximate nearest neighbor (ANN) search [REF3]. These models have shown state-of-the-art accuracy and behave differently from classic retrieval methods. Dense retrieval models leverage the representations of documents and queries to perform retrieval, without relying on traditional scoring functions like BM25. Recent advancements in dense retrieval have led to the development of new search systems [REF3].

In addition to the aforementioned approaches, the selection of negative examples plays a crucial role in learning high-quality encoders [REF5]. Negative examples are often selected from a large pool, and different types of negatives, such as random, BM25, and gold passages, have been considered. The impact of different types of negative passages and training schemes on retrieval performance has been studied [REF5].

Overall, representation-focused systems that utilize single representations have shown promising results in improving retrieval performance. Passage selection models, dynamic hard negatives, and dense retrieval models are some of the approaches that leverage single representations to enhance retrieval accuracy. The selection of negative examples also plays a crucial role in training high-quality encoders. These advancements in representation-focused systems contribute to the development of more effective and efficient neural information retrieval systems.

References sent to GTP:

REF0 - Dense Passage Retrieval for Open-Domain Question Answering

REF1 - Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval

REF2 - Efficient Document Re-Ranking for Transformers by Precomputing Term Representations

REF3 - Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval

REF4 - Optimizing Dense Retrieval Model Training with Hard Negatives

REF5 - Dense Passage Retrieval for Open-Domain Question Answering

REF6 - Dense Passage Retrieval for Open-Domain Question Answering

REF7 - Optimizing Dense Retrieval Model Training with Hard Negatives

REF8 - Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval

REF9 - Dense Passage Retrieval for Open-Domain Question Answering

3.2 Multiple Representations

Representation-focused Systems - Multiple Representations

In neural information retrieval, representation-focused systems aim to improve retrieval performance by leveraging multiple representations of queries and documents. These systems utilize neural networks to learn dense representations that capture the semantic meaning of text, enabling more effective matching and ranking of relevant documents. This section explores the use of multiple representations in representation-focused systems, highlighting their benefits and discussing relevant approaches.

One common approach in representation-focused systems is to use learned neural networks, such as Deep LM (Language Model) and BERT (Bidirectional Encoder Representations from Transformers), to generate dense representations of queries and documents [REF0]. These representations are then used to compute relevance scores, which determine the ranking of documents. For example, BERT's [CLS] output token is often used to produce a relevance score [REF0]. This approach has achieved state-of-the-art reranking performance for passages and documents [REF0].

To further improve dense retrieval systems, researchers have explored the use of better training techniques and extended single vector systems to multi-vector representation systems [REF1]. Polyencoder encodes queries into a set of vectors, while Me-BERT represents documents with a set of vectors [REF1]. ColBERT, on the other hand, uses multiple vectors to encode both queries and documents, computing all-to-all soft matches between tokens [REF1]. These multi-vector representation systems have shown promising results in enhancing retrieval performance [REF1].

While learned dense representations have shown significant advancements in neural information retrieval, it is important to note that they may not always outperform sparse features, particularly in tasks that require precise detection of word overlap [REF2]. For instance, the BM25 model has been shown to outperform a dual encoder based on BERT, especially on longer documents [REF2]. This raises questions about the limitations of dual encoders and the circumstances in which they may not reach the state of the art [REF2].

To address the limitations of exact match scoring, researchers have proposed modifying the exact lexical match framework with contextualized term representations [REF4]. Instead of relying solely on term frequency, these systems encode the semantics of terms using contextualized vector representations and perform matching between exact lexical matched tokens [REF4]. This modification allows for more effective and efficient retrieval systems [REF4].

Another important aspect of representation-focused systems is the interaction mechanism between queries and documents. While more sophisticated matching mechanisms are possible, a summation of maximum similarity computations has been found to be particularly efficient and amenable to pruning for top-k retrieval [REF5]. This approach enables skipping documents without materializing the full interaction matrix, resulting in improved efficiency [REF5].

In the context of neural information retrieval, the use of bi-encoder and cross-encoder architectures has been explored [REF7]. Bi-encoder methods cache the representations of a large candidate set, making them efficient during evaluation [REF7]. Cross-encoders, on the other hand, make no assumptions on the similarity scoring function between input and label, allowing for more flexible matching based on any dependencies [REF7]. These architectures have been applied in various models, including Memory Networks, Transformers, LSTMs, CNNs, and self-attention-based architectures [REF7].

In conclusion, representation-focused systems in neural information retrieval leverage multiple representations of queries and documents to improve retrieval performance. These systems utilize learned neural networks to generate dense representations, enabling more effective matching and ranking. The use of multi-vector representations, modifications to exact lexical match frameworks, and efficient interaction mechanisms contribute to the advancements in representation-focused systems.

References sent to GTP:

REF0 - COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List

REF1 - COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List

REF2 - Sparse, Dense, and Attentional Representations for Text Retrieval

REF3 - Sparse, Dense, and Attentional Representations for Text Retrieval

REF4 - COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List

REF5 - ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

REF6 - ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

REF7 - Real-time Inference in Multi-sentence with Deep Pretrained Transformers

REF8 - ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

REF9 - Real-time Inference in Multi-sentence with Deep Pretrained Transformers

3.3 Fine-tuning Representation-focused Systems

Fine-tuning representation-focused systems is an essential aspect of neural information retrieval. Fine-tuning allows us to adapt pre-trained models to specific retrieval tasks, improving their performance and effectiveness. In this section, we discuss the process of fine-tuning representation-focused systems and its impact on retrieval accuracy.

One approach to fine-tuning representation-focused systems is to train them using a limited number of question-passage pairs. For instance, it has been shown that a high-quality dense retriever can be trained using as few as 1,000 examples, outperforming traditional retrieval methods like BM25 [REF2]. This suggests that with a general pre-trained language model, it is possible to achieve impressive retrieval accuracy even with a small training set.

In addition to the number of training examples, the choice of negative passages during training also plays a role in fine-tuning representation-focused systems. In the standard 1-of-N training setting, each question in the batch is paired with a positive passage and a set of negative passages [REF3]. Interestingly, the choice of negatives, whether random, BM25, or gold passages, does not significantly impact the top-k accuracy when k is greater than or equal to 20 [REF3]. However, in-batch negative training, where gold negative passages come from the same batch, has been shown to substantially improve retrieval results [REF3]. This approach allows for the reuse of negative examples already in the batch, increasing the number of training instances and contributing to better model performance [REF3].

Another aspect of fine-tuning representation-focused systems is the combination of different retrieval methods. For example, combining the scores of BM25 and a dense passage retriever (DPR) using a linear combination has been shown to improve retrieval accuracy [REF6]. By obtaining two initial sets of top-2000 passages based on BM25 and DPR, respectively, and reranking their union, a more effective ranking function can be achieved [REF6].

Fine-tuning representation-focused systems also involves training the question and passage encoders for multiple epochs using techniques such as linear scheduling with warm-up and dropout regularization [REF6]. While it is desirable to have a single retriever that performs well across different datasets, training a multi-dataset encoder by combining training data from multiple datasets has shown promising results [REF6]. Additionally, incorporating clickthrough data for model training has been found to significantly improve retrieval performance [REF4].

In summary, fine-tuning representation-focused systems involves training them with a limited number of question-passage pairs, carefully selecting negative passages, combining different retrieval methods, and optimizing training parameters. These approaches have shown to enhance retrieval accuracy and contribute to the effectiveness of neural information retrieval systems.

References sent to GTP:

REF0 - Dense Passage Retrieval for Open-Domain Question Answering

REF1 - Dense Passage Retrieval for Open-Domain Question Answering

REF2 - Dense Passage Retrieval for Open-Domain Question Answering

REF3 - Dense Passage Retrieval for Open-Domain Question Answering

REF4 - Learning Deep Structured Semantic Models for Web Search using Clickthrough Data

REF5 - Dense Passage Retrieval for Open-Domain Question Answering

REF6 - Dense Passage Retrieval for Open-Domain Question Answering

REF7 - Dense Passage Retrieval for Open-Domain Question Answering

REF8 - Dense Passage Retrieval for Open-Domain Question Answering

REF9 - Dense Passage Retrieval for Open-Domain Question Answering

4 Retrieval Architectures and Vector Search

4.1 MIP and NN Search Problems

Retrieval Architectures and Vector Search - MIP and NN Search Problems

In the context of neural information retrieval, retrieval architectures and vector search play a crucial role in efficiently finding approximate solutions for Maximum Inner Product Search (MIPS) problems. MIPS problems involve finding a data vector from a collection of "database" vectors that maximizes the inner product with a given query vector [REF7]. To address this problem, Shrivastava and Li (2014a) propose constructing a Locality Sensitive Hash (LSH) for inner product similarity [REF7].

LSH, originally introduced by Indyk and Motwani (1998), is a widely used tool for approximate nearest neighbor search [REF7]. It has also found applications in various other domains [REF7]. In the case of MIPS, LSH can be used to efficiently find approximate solutions by mapping vectors into hash codes and comparing the hash codes of the query and database vectors [REF7].

One approach to LSH for MIPS is the L2-ALSH(SL) method proposed by Shrivastava and Li (2014a) [REF7]. L2-ALSH(SL) utilizes a pair of mappings, P(x) and Q(y), to transform the vectors into a joint space Z [REF1]. The L2-ALSH(SL) method employs a standard L2 hash function to generate hash codes for the transformed vectors [REF1]. However, it has been shown that L2-ALSH(SL) does not perform optimally in terms of hashing quality [REF4] and requires tuning of parameters specific to the dataset [REF6].

To address the limitations of L2-ALSH(SL), Shrivastava and Li (2014b) proposed a modified hash function called SIGN-ALSH(SL) [REF4]. SIGN-ALSH(SL) is based on random projections and incorporates an asymmetric transform similar to L2-ALSH(SL) [REF4]. However, even though SIGN-ALSH(SL) shows improvement over L2-ALSH(SL), it still requires parameter tuning and is not universally applicable [REF6].

In contrast to L2-ALSH(SL) and SIGN-ALSH(SL), SIMPLE-LSH is a parameter-free and universal hashing method for MIPS [REF6]. SIMPLE-LSH does not require any parameter tuning and achieves superior empirical performance compared to L2-ALSH(SL) [REF4][REF5]. The simplicity and effectiveness of SIMPLE-LSH make it a promising approach for vector search in neural information retrieval tasks.

In conclusion, retrieval architectures and vector search methods, such as L2-ALSH(SL), SIGN-ALSH(SL), and SIMPLE-LSH, play a crucial role in addressing the MIPS problem in neural information retrieval. While L2-ALSH(SL) and SIGN-ALSH(SL) have been proposed as hash functions for MIPS, they have limitations in terms of parameter tuning and hashing quality. On the other hand, SIMPLE-LSH offers a parameter-free and universal approach that shows superior empirical performance. These methods provide valuable tools for efficient and accurate vector search in neural information retrieval systems.

References sent to GTP:

REF0 - On Symmetric and Asymmetric LSHs for Inner Product Search

REF1 - On Symmetric and Asymmetric LSHs for Inner Product Search

REF2 - On Symmetric and Asymmetric LSHs for Inner Product Search

REF3 - On Symmetric and Asymmetric LSHs for Inner Product Search

REF4 - On Symmetric and Asymmetric LSHs for Inner Product Search

REF5 - On Symmetric and Asymmetric LSHs for Inner Product Search

REF6 - On Symmetric and Asymmetric LSHs for Inner Product Search

REF7 - On Symmetric and Asymmetric LSHs for Inner Product Search

REF8 - On Symmetric and Asymmetric LSHs for Inner Product Search

REF9 - On Symmetric and Asymmetric LSHs for Inner Product Search

4.2 Locality sensitive hashing approaches

Retrieval Architectures and Vector Search - Locality Sensitive Hashing Approaches

Locality Sensitive Hashing (LSH) is a technique used in retrieval architectures and vector search to efficiently perform similarity search on large-scale datasets. LSH utilizes hash functions that map similar objects into the same hash buckets with a high probability [REF8]. This approach consists of two steps: selecting "candidate" objects using LSH functions for a given query, and ranking the candidate objects based on their distances to the query [REF8].

LSH has been widely applied in various domains, including data compression, databases, information retrieval, image and video databases, machine learning, pattern recognition, and statistics [REF5]. Initially, LSH was primarily used for binary data in the Hamming space [REF2]. However, it has been extended to work directly on points in Euclidean space without embeddings [REF2]. This extension allows LSH to handle high-dimensional datasets, which is crucial considering the "curse of dimensionality" [REF5].

One important aspect of LSH is the evaluation of its performance. Evaluation benchmarks are created by randomly selecting query objects and defining their ground truth as the K nearest neighbors based on the Euclidean distance of their feature vectors [REF0]. The performance of a similarity search system is measured in terms of search quality, search speed, and space requirement [REF0]. Recall, which measures the proportion of correctly retrieved neighbors, is commonly used to evaluate search quality [REF0]. Ideally, a similarity search system should achieve high-quality search with high speed while using minimal space [REF0].

To improve the efficiency of LSH, various approaches have been proposed. The entropy-based approach aims to reduce duplicate buckets and improve search speed [REF1]. By analyzing the differences between query-directed and step-wise probing sequences, it has been shown that the query-directed probing sequence requires significantly fewer hash tables while achieving similar query times [REF1]. Additionally, a refined construction of a probing sequence can take advantage of the position of the query within its slot, leading to more efficient perturbations worth considering [REF7].

In summary, LSH is a powerful technique for retrieval architectures and vector search, allowing efficient similarity search in large-scale datasets. It has been extended to handle high-dimensional Euclidean space and has been applied in various domains. Evaluation benchmarks and metrics, such as recall, are used to assess the performance of LSH-based systems. The entropy-based approach and query-directed probing sequences are among the methods used to improve the efficiency of LSH.

References sent to GTP:

REF0 - Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search

REF1 - Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search

REF2 - Locality-Sensitive Hashing Scheme Based on p-Stable Distributions

REF3 - Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search

REF4 - Locality-Sensitive Hashing Scheme Based on p-Stable Distributions

REF5 - Locality-Sensitive Hashing Scheme Based on p-Stable Distributions

REF6 - Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search

REF7 - Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search

REF8 - Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search

REF9 - Locality-Sensitive Hashing Scheme Based on p-Stable Distributions

4.3 Vector quantisation approaches

Retrieval Architectures and Vector Search - Vector quantisation approaches

Vector quantisation is a widely used technique in information retrieval systems for efficient and accurate retrieval of relevant documents. In this section, we will discuss retrieval architectures and vector search, specifically focusing on vector quantisation approaches.

One approach that has been explored is the use of Hamming embedding methods [REF0]. These methods utilize the Hamming space, which allows for a limited number of distinct distances. However, this limitation can be overcome by using table lookups to compute Hamming distances efficiently. By employing a modified inverted file structure, the most relevant vectors can be accessed rapidly, avoiding the need for exhaustive comparison with all codes [REF0].

Another approach is product quantization, which has shown significant improvements in approximate nearest neighbor search [REF1]. This technique involves dividing the input vector into subvectors and quantizing each subvector separately. The resulting compact coding scheme provides an accurate approximation of the Euclidean distance. Additionally, the use of an inverted file system further enhances efficiency by avoiding exhaustive search [REF3]. Experimental results have demonstrated the superiority of product quantization in terms of search quality and memory usage [REF3].

Trade-offs between code length and search quality have also been investigated [REF4]. The product quantizer is parametrized by the number of subvectors and the number of quantizers per subvector, which determines the code length. By varying these parameters, the trade-off between code length and search quality can be analyzed. This analysis has shown that higher values of the number of quantizers generally lead to better efficiency [REF4].

The impact of component grouping in vector quantization has also been studied [REF9]. In structured vectors such as SIFT and GIST descriptors, which are composed of concatenated orientation histograms, the bins of a histogram may end up in different quantization groups when using a product quantizer. By considering the prior knowledge of the descriptor design and grouping the components accordingly, further improvements in search efficiency can be achieved [REF9].

In conclusion, retrieval architectures and vector search play a crucial role in neural information retrieval. Vector quantisation approaches, such as Hamming embedding methods and product quantization, have shown promising results in terms of efficiency and accuracy. By considering trade-offs, such as code length and search quality, and incorporating prior knowledge of the data structure, further advancements can be made in the field of neural information retrieval.

References sent to GTP:

REF0 - Product Quantization for Nearest Neighbor Search

REF1 - Product Quantization for Nearest Neighbor Search

REF2 - Vector Quantization and Signal Compression

REF3 - Product Quantization for Nearest Neighbor Search

REF4 - Product Quantization for Nearest Neighbor Search

REF5 - Product Quantization for Nearest Neighbor Search

REF6 - Product Quantization for Nearest Neighbor Search

REF7 - Product Quantization for Nearest Neighbor Search

REF8 - Product Quantization for Nearest Neighbor Search

REF9 - Product Quantization for Nearest Neighbor Search

4.4 Graph approaches

Retrieval Architectures and Vector Search - Graph approaches

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

One popular graph-based retrieval architecture is the NN-Descent algorithm [REF0]. This algorithm utilizes the observation that exploring the neighbors' neighbors of a data point can lead to the discovery of its nearest neighbors. By iteratively applying this observation, NN-Descent efficiently identifies the nearest neighbors of each object in the dataset. The algorithm starts with a random K-nearest neighbor graph approximation and progressively refines it to improve the accuracy of the retrieved nearest neighbors.

To evaluate the performance of graph-based retrieval architectures, various metrics are used. Recall, which measures the fraction of true results within a search, is commonly employed [REF2]. Additionally, the fraction of visited elements during a search is calculated to assess the complexity of the search process [REF2]. Empirical studies have been conducted to analyze the impact of different parameters, such as the dimensionality of the data, on the performance of graph-based retrieval architectures [REF6]. These studies provide insights into selecting suitable parameter values to optimize retrieval performance.

Efficient construction of K-Nearest Neighbor Graphs (K-NNGs) is a crucial aspect of graph-based retrieval architectures. Paredes et al. proposed two methods for constructing K-NNGs in general metric spaces with low empirical complexity [REF3]. However, these methods require a global data structure and are challenging to parallelize across machines. To address this issue, efficient methods based on recursive data partitioning and space-filling curves have been developed for Euclidean spaces [REF3]. Nonetheless, these methods do not naturally generalize to other distance metrics or similarity measures.

The development of fast and scalable algorithms for K-Nearest Neighbor Search (K-NNS) has been a topic of great interest [REF4]. The naïve approach of computing distances between the query and every element in the dataset is infeasible for large-scale datasets due to its linear complexity. Approximate Nearest Neighbor Search (K-ANNS) algorithms have been proposed to overcome this limitation by allowing a small number of errors in the search process [REF4]. These algorithms relax the condition of exact search and offer significant speedup in search operations.

In recent advancements, modifications have been made to the k-NN algorithm to improve its performance [REF5]. These modifications include using a different stop condition and sharing the list of previously visited elements across multiple searches. By exploring the neighborhood of the closest elements in a greedy manner, the algorithm iteratively improves the known k closest elements. This approach reduces unnecessary repeated extractions and enhances the efficiency of the search process.

In conclusion, graph-based approaches have emerged as powerful tools for neural information retrieval. Retrieval architectures utilizing graph structures, such as the NN-Descent algorithm, enable efficient identification of nearest neighbors. Various metrics, including recall and the fraction of visited elements, are used to evaluate the performance of these architectures. Efficient construction of K-NNGs and the development of fast and scalable K-NNS algorithms are ongoing research areas. Modifications to the k-NN algorithm have also been proposed to enhance its performance. These advancements contribute to the effective retrieval of relevant information in neural information retrieval systems.

References sent to GTP:

REF0 - Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures

REF1 - Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures

REF2 - Approximate Nearest Neighbor Algorithm based on Navigable Small World Graphs

REF3 - Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures

REF4 - Efficient and Robust Approximate Nearest Neighbor Search using Hierarchical Navigable Small World Graphs

REF5 - Approximate Nearest Neighbor Algorithm based on Navigable Small World Graphs

REF6 - Efficient K-Nearest Neighbor Graph Construction for Generic Similarity Measures

REF7 - The Small-World Phenomenon - An Algorithmic Perspective

REF8 - Approximate Nearest Neighbor Algorithm based on Navigable Small World Graphs

REF9 - Efficient and Robust Approximate Nearest Neighbor Search using Hierarchical Navigable Small World Graphs

4.5 Optimisations

Retrieval Architectures and Vector Search - Optimisations

In neural information retrieval, retrieval architectures and vector search play a crucial role in optimizing the efficiency and effectiveness of the retrieval process. This section discusses various techniques and approaches used to enhance retrieval architectures and improve vector search performance.

One common approach is to use the output embedding of a special token, such as the "[CLS]" token, as the representation of the input text [REF0]. This allows for efficient computation of relevance scores using inner product similarity. To perform efficient similarity search, libraries like Faiss are often adopted [REF0]. Additionally, document truncation is employed to limit the length of documents, with a maximum of 120 tokens for passage tasks and 512 tokens for document tasks [REF0].

To improve retrieval performance, the use of hard negatives is a common strategy. For instance, the top-200 documents can be selected as hard negatives [REF0]. Training models on passage tasks with specific optimization techniques, such as the Lamb optimizer, batch size of 256, and learning rate of 2 × 10−4, can also contribute to better retrieval architectures [REF0].

In terms of efficiency, it is important to consider the trade-off between effectiveness and latency. Complex end-to-end neural retrieval models may achieve better ranking performance but often suffer from higher query latency due to their intricate architectures [REF1]. To address this, a reranking stage can be added to retrieval architectures like RepCONC, allowing for a comparison in terms of effectiveness-latency tradeoff [REF1].

Another aspect to consider is the impact of query token frequency on retrieval effectiveness. Query embedding pruning strategies based on the collection frequency of query tokens have been proposed, showing that lower frequency tokens can be more discriminative and retrieve relevant documents effectively [REF2]. By utilizing the importance of query embeddings based on token frequency, retrieval architectures can achieve comparable effectiveness results with a reduced number of query embeddings [REF2].

In the context of brute-force document retrieval (DR) models, which retrieve candidates for queries through exhaustive search, various architectures have been explored [REF3]. These models share similar architectures but differ in their training processes. Baselines trained using negative sampling methods, such as Rand Neg, BM25 Neg, ANCE, and STAR, have been widely studied [REF3].

Efficient vector search is often achieved by leveraging the capabilities of GPUs. Vector representations, such as image descriptors or embeddings, are typically high-dimensional vectors that require GPU systems for effective production and manipulation [REF6]. Utilizing GPUs for similarity search can significantly improve the speed achievable on a single machine [REF4]. Constructing k-nearest neighbor graphs of datasets via brute force is one example of utilizing similarity search methods [REF4].

Comparisons between different retrieval models are also essential. Traditional bag-of-words models, such as doc2query, DeepCT, and docTTTTquery, rely on techniques like term re-weighing and term expansion to enhance retrieval effectiveness [REF5]. Latency comparisons between these models can be conducted using implementations like Anserini [REF5].

In the context of late interaction, relevance scores between queries and documents are estimated based on the interaction between their bags of contextualized embeddings [REF7]. This approach allows for the computation of relevance scores while filtering out unnecessary embeddings, such as those corresponding to punctuation symbols [REF7].

Overall, retrieval architectures and vector search optimizations play a crucial role in enhancing the efficiency and effectiveness of neural information retrieval systems. Techniques such as utilizing special tokens, employing hard negatives, considering token frequency, and leveraging GPU capabilities contribute to improved retrieval performance. Comparisons between different retrieval models and the exploration of late interaction approaches further advance the field of neural information retrieval.

References sent to GTP:

REF0 - Optimizing Dense Retrieval Model Training with Hard Negatives

REF1 - Learning Discrete Representations via Constrained Clustering for Effective and Efficient Dense Retrieval

REF2 - Query Embedding Pruning for Dense Retrieval

REF3 - Jointly Optimizing Query Encoder and Product Quantization to Improve Retrieval Performance

REF4 - Billion-Scale Similarity Search with GPUs

REF5 - ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

REF6 - Billion-Scale Similarity Search with GPUs

REF7 - ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT

REF8 - Learning Discrete Representations via Constrained Clustering for Effective and Efficient Dense Retrieval

REF9 - Jointly Optimizing Query Encoder and Product Quantization to Improve Retrieval Performance

5 Learned Sparse Retrieval

5.1 Document expansion learning

Learned Sparse Retrieval - Document expansion learning

Document expansion learning is a technique in neural information retrieval that aims to improve retrieval performance by expanding the original query with additional relevant terms. This approach recognizes that the original query may not capture all the necessary information needed to retrieve relevant documents accurately. By incorporating additional terms, document expansion learning aims to enhance the retrieval process and increase the likelihood of retrieving relevant documents.

One key aspect of document expansion learning is the understanding of term mismatch, which plays a central role in retrieval theory [REF0]. Term mismatch refers to the situation where the query term and the relevant document do not perfectly match, leading to potential retrieval challenges. The theoretical role and practical significance of term mismatch have been clarified, shedding light on the behaviors of current retrieval models and techniques [REF0].

The quantification of term mismatch in retrieval and the analysis of its variation and causes have enabled researchers to design novel methods for predicting term mismatch [REF0] [REF2]. These prediction methods utilize query-dependent features to accurately estimate the likelihood of term mismatch for a given query [REF0]. By incorporating these predictions into the retrieval process, ad hoc retrieval can be improved through term weighting based on mismatch predictions [REF0].

Query expansion, particularly in the Conjunctive Normal Form (CNF), has been highlighted as an important approach for addressing term mismatch [REF0]. CNF query expansion allows for the inclusion of additional terms that can help resolve the mismatch between the query and relevant documents [REF0]. Furthermore, document expansion learning leverages term mismatch as a diagnostic tool to improve the effectiveness of query expansion [REF0].

To facilitate document expansion learning, various techniques have been proposed. One approach involves the use of latent semantic analysis, such as local Latent Semantic Indexing (LSI), to generate query-dependent features for predicting term mismatch [REF4]. This approach considers the different senses and synonyms of query terms, allowing for more accurate predictions of term mismatch [REF4].

Another approach utilizes instance-based learning frameworks to predict the variation of term recall for the same term across different queries [REF3]. By leveraging historic occurrences of a test term and their predicted recall differences, a single prediction of term recall can be generated for a new query [REF3]. This approach enables efficient and effective prediction of term recall without the need for multiple passes over the data [REF3].

In conclusion, document expansion learning is a valuable technique in neural information retrieval that aims to improve retrieval performance by expanding the original query with additional relevant terms. By understanding and predicting term mismatch, as well as leveraging query expansion and instance-based learning frameworks, document expansion learning enhances the retrieval process and increases the likelihood of retrieving relevant documents.

References sent to GTP:

REF0 - Modeling and Solving Term Mismatch for Full-Text Retrieval

REF1 - Modeling and Solving Term Mismatch for Full-Text Retrieval

REF2 - Modeling and Solving Term Mismatch for Full-Text Retrieval

REF3 - Modeling and Solving Term Mismatch for Full-Text Retrieval

REF4 - Modeling and Solving Term Mismatch for Full-Text Retrieval

REF5 - Modeling and Solving Term Mismatch for Full-Text Retrieval

REF6 - Modeling and Solving Term Mismatch for Full-Text Retrieval

REF7 - Modeling and Solving Term Mismatch for Full-Text Retrieval

REF8 - Modeling and Solving Term Mismatch for Full-Text Retrieval

REF9 - Modeling and Solving Term Mismatch for Full-Text Retrieval

5.2 Impact score learning

Learned Sparse Retrieval - Impact score learning

In the field of neural information retrieval, one important aspect is the scoring of documents to determine their relevance to a given query. Traditional retrieval models often rely on pre-defined scoring functions, such as language modeling or BM25, which may not capture the complex relationships between queries and documents. To address this limitation, learned sparse retrieval techniques have emerged, which aim to learn the impact scores of query terms on document relevance.

The concept of impact score learning involves training a model to assign impact scores to query terms based on their importance in determining document relevance. These impact scores can then be used to rank and retrieve documents more effectively. Several approaches have been proposed in the literature to learn impact scores, and we will discuss some of them in this section.

One approach to impact score learning is to use regression or decision trees to model the relationship between query terms and document relevance [REF5]. In this approach, a set of regression or decision trees is trained using features derived from the query and document. Each tree assigns a weight to each query term based on its impact on document relevance. The impact scores of query terms are then computed as the weighted sum of the tree scores. This approach has been shown to be effective in capturing the complex interactions between query terms and document relevance.

Another approach to impact score learning is based on the cascading nature of web search [REF2]. In this approach, different ranking features, such as PageRank or URL information, are calculated and used within a learned model to re-rank the documents. The impact scores of query terms are learned based on their contribution to the final ranking of documents. This approach takes into account various factors that can influence document relevance and provides a more comprehensive view of the impact of query terms.

Efficiency is a crucial aspect of information retrieval systems, and learned sparse retrieval techniques also consider this aspect. Pruning strategies, such as block-based dynamic pruning, can be applied to reduce the computational cost of impact score learning [REF6]. These strategies aim to avoid scoring documents that are unlikely to be in the top-ranked results. By dynamically pruning the search space, the computational overhead can be significantly reduced without sacrificing retrieval effectiveness.

In summary, learned sparse retrieval techniques offer a promising approach to improve the effectiveness and efficiency of information retrieval systems. By learning the impact scores of query terms, these techniques can better capture the complex relationships between queries and documents. Regression or decision trees, as well as cascading models, have been proposed to learn impact scores. Additionally, pruning strategies can be applied to optimize the computational cost of impact score learning. These techniques pave the way for more advanced and accurate retrieval models in neural information retrieval.

[REF2] Blanco, R., Catena, M., & Tonellotto, N. (2016). Exploiting Green Energy to Reduce the Operational Costs of Multi-Center Web Search Engines. In Proceedings of the 25th International Conference on World Wide Web (WWW '16).

[REF5] Lin, J., & Trotman, A. (2017). The role of index compression in score-at-a-time query evaluation. Information Retrieval Journal, 1-22.

[REF6] Ling, X., Deng, W., Gu, C., Zhou, H., Li, C., & Sun, F. (2017). Model Ensemble for Click Prediction in Bing Search Ads.

[REF9] Anh, V. N., & Moffat, A. (1998). Compressed Inverted Files with Reduced Decoding Overheads. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98).

References sent to GTP:

REF0 - Efficient Query Processing for Scalable Web Search

REF1 - Efficient Query Processing for Scalable Web Search

REF2 - Efficient Query Processing for Scalable Web Search

REF3 - Efficient Query Processing for Scalable Web Search

REF4 - Efficient Query Processing for Scalable Web Search

REF5 - Efficient Query Processing for Scalable Web Search

REF6 - Efficient Query Processing for Scalable Web Search

REF7 - Efficient Query Processing for Scalable Web Search

REF8 - Efficient Query Processing for Scalable Web Search

REF9 - Efficient Query Processing for Scalable Web Search

5.3 Sparse representation learning

Learned Sparse Retrieval - Sparse representation learning

Efficient approximate Nearest-Neighbor Search (NNS) and Maximum Inner-Product Search (MIPS) are active areas of research in the field of neural information retrieval [REF0]. To address the challenge of accurate search in high dimensions, researchers have focused on learning compact lower-dimensional representations that preserve distance information [REF0]. However, the exploration of learning sparse higher-dimensional representations has only been recently addressed [REF0].

Jeong and Song (2018) propose an end-to-end approach to learn sparse and high-dimensional hashes, which leads to significant speed-up in retrieval time compared to dense embeddings [REF0]. Their approach demonstrates the potential of learning sparse representations for efficient information retrieval. Similarly, Cao et al. (2018) also explore the learning of sparse representations, highlighting the benefits of sparsity in retrieval tasks [REF0].

Sparse representation learning offers several advantages in neural information retrieval. By learning sparse representations, it becomes possible to reduce the computational cost associated with high-dimensional search spaces [REF0]. This is particularly important in scenarios where accurate search in high dimensions is prohibitively expensive [REF0]. Sparse representations also enable efficient approximate NNS and MIPS, striking a balance between accuracy and efficiency [REF0].

In the context of learned sparse retrieval, the estimation of activation probabilities for tokens in documents or queries plays a crucial role [REF1]. These probabilities are empirically estimated from a set of development queries, allowing for the prediction of term importance [REF1]. The importance of terms is then used to construct sparse representations, which can outperform other sparse retrieval methods and achieve competitive results with state-of-the-art dense retrieval methods [REF1].

Analyzing the expansion of terms in learned sparse retrieval models provides insights into the effectiveness of these approaches [REF2]. The expanded terms, predicted by the Gating Controller, demonstrate the ability of the model to activate important terms that may not appear in the original passage or query [REF2]. The expanded terms can be categorized into passage-to-query terms, synonyms of original terms, and co-occurred words for the original terms [REF2]. This analysis highlights the capability of learned sparse retrieval models to capture semantic similarities and improve retrieval performance [REF2].

Sparse representation learning also involves the optimization of the expected Floating Point Operations (FLOPs) [REF3]. Since FLOPs are a discontinuous function of model parameters, optimizing them directly is challenging. Instead, a continuous relaxation of FLOPs is used, and a regularized loss function is employed to minimize the loss while controlling the expected FLOPs [REF3]. This approach allows for the efficient optimization of sparse representation models [REF3].

It is important to note that FLOPs reduction may not always accurately reflect the actual speedup on mainstream commercial processors [REF4]. While FLOPs reduction is a reasonable measure of speedup on processors with limited parallelization and cache memory, it may not capture the optimized cache and SIMD mechanisms of mainstream processors [REF4]. However, research on hardware architectures tailored to sparse operations has shown promising speedup proportional to FLOPs reduction [REF4]. Further exploration of hardware aspects and their impact on performance is an avenue for future research [REF4].

In summary, learned sparse retrieval and sparse representation learning offer efficient and effective solutions for information retrieval tasks. By learning compact and sparse representations, these approaches enable accurate and efficient search in high-dimensional spaces. The estimation of activation probabilities, analysis of term expansion, and optimization of FLOPs contribute to the success of learned sparse retrieval models. Future research can focus on further improving the interpretability and performance of these models in various retrieval scenarios.

References sent to GTP:

REF0 - Minimizing FLOPS to Learn Efficient Sparse Representations

REF1 - SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking

REF2 - SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval

REF3 - Minimizing FLOPS to Learn Efficient Sparse Representations

REF4 - Minimizing FLOPS to Learn Efficient Sparse Representations

REF5 - SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking

REF6 - SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval

REF7 - SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval

REF8 - SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking

REF9 - SparTerm: Learning Term-based Sparse Representation for Fast Text Retrieval