COL 764 - Final Examination

17th November 2024

This is closed book exam. It is **NOT** a collaborative exam. You are expected not to seek help from any other person(s)/resource(s) during the exam.

Total time: 2 hours **Total Points: 40**

Name: ARCHISMAN BISWAS

Entrynumber: 2021MT10254

Signature:

Date: 17/11/2024

Instructions

- Answer all questions.
- Fill your **entry number in every page** in the space given.
- No additional sheets will be provided.
- **All answers must be provided in the space given immediately after the question.** If you need to correct an answer, then you must *cleanly cross-out* the incorrect answer you have filled, and write it separately at the end (write the question number correctly).
- While every effort is made to minimize ambiguity in the questions, it is possible that some may find certain questions unclear. In such a situation, you must make appropriate and meaningful assumptions, state them clearly before answering the question. No clarifications will be provided during the exam.
- For multiple choice questions, there may be more than one option that is correct. Your answer is correct *only if all and only correct options* are marked.

(Questions Begin in Next Page)

1. (1 point) Which of the following is aimed towards reducing the cognitive overhead on users of Web search systems:
   A. Result ranking based on their estimated relevance score
   B. Efficient retrieval
   C. Massive (Web-scale) collections
   D. Preview generation for each result generated
   E. Advertisements

   1. AD

2. (1 point) While using the relevance scores of different systems for developing an aggregate ranking, one of the oft-encountered problem is that these scores are not comparable across systems. Thus, we need to normalize the scores before we can start aggregating. Which of the following is a desirable property of such a normalization:
   A. It should be resilient to addition of an arbitrary constant to the scores.
   B. It should be resilient to addition of an arbitrary constant to the variance of the distribution of retrieved scores.
   C. It should be resilient to multiplication of scores by an arbitrary constant.
   D. It should be able to map all score distributions to a normal distribution with appropriate mean and variance.
   E. none of the above.

   2. AC

3. (1 point) Which of the following are the possible sources of unfairness in information retrieval?
   A. Biases in training data while learning to rank
   B. Biases in word usage distribution in the corpus we are retrieving from
   C. Biases in the signals to capture user preferences
   D. Biases in document content structure
   E. None of the above

   3. ABC

4. (1 point) Which of the following is/are true?
   A. Aggregating passage representations is more effective than aggregating passage scores while trying to rank multi-passage documents using BERT.
   B. Training with term replacement objective instead of random masking during BERT provides improved effectiveness in retrieval.
   C. BERT based reranking starts performing poorly when the number of candidate passages from the initial round of retrieval increases.
   D. monoBERT ranking is a pointwise learning to rank approach.

   4. AB D

5. One of the successful learning to rank technique is that of RankSVM or Ranking SVM (which we studied in our course). The SVM formulation is given as below:

$$\text{minimize } V(\vec{w}, \vec{\xi}) = \frac{1}{2}\vec{w} \cdot \vec{w} + C \sum \xi_{i,j,k}$$

subject to:

$$\forall i \forall j \forall k : \xi_{i,j,k} \geq 0$$
$$\forall (d_i, d_j) \in r_1^* : \vec{w}\Phi(q_1, d_i) > \vec{w}\Phi(q_1, d_j) + 1 - \xi_{i,j,1}$$
$$\dots$$
$$\dots$$
$$\forall (d_i, d_j) \in r_n^* : \vec{w}\Phi(q_n, d_i) > \vec{w}\Phi(q_n, d_j) + 1 - \xi_{i,j,n}$$

Based on this, answer:

(a) (1 point) What does r_k^* represent (in words)?

r_k^* represents the ~~optimal ranking~~ given complete ordering of document preferences corresponding to the query q_k, which is a part of the training data

In other words, tuples (q_i, r_i^*) are given to us during training, where r_i^* refers to the true ~~ranking~~ ordering (complete) over docs

(b) (1 point) What are $\xi_{i,j,k}$ (in words)?

$\xi_{i,j,k}$ are the terms used ~~as~~ for regularization. It is the slack which is allowed at the boundary.

(c) (1 point) What does $(d_i, d_j) \in r_k^*$ mean (in words)? How do you obtain this information during training?

r_k^* is the true ~~ranking~~ ordering which is given during training

So, $(d_i, d_j) \in r_k^*$ means that in r_k^* d_i is ranked ~~over~~ above d_j

During training, r_k^* is given to us, so we can take any pair of documents and check if it belongs to r_k^*

(Continued...)

(d) (2 points) What is $\Phi(q_k, d_i)$ (in words)? Illustrate with an example.

Φ measures the similarity between two vectors.
cosine similarity is an example where

$$\varphi(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|}$$

(e) (2 points) Briefly explain what is being learnt, and how does one use it during query processing (i.e., inference/testing) stage of the search engine?

Here, the weight vector $\vec{w}$ & the variables $\xi_{i,j,k} \forall i,j,k$ are being learnt. These would explain the boundaries which would classify each document pair as more or less preferred

During inference, we will have an ordering over the set of documents we use this to achieve a ranking.

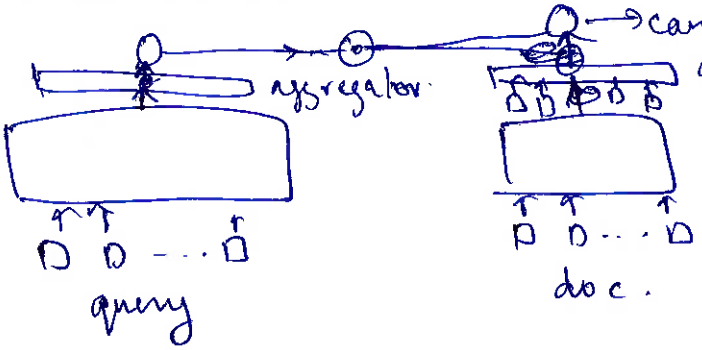
6. (a) (2 points) What are the main limitations of using the pretrained BERT model and embeddings for general purpose information retrieval tasks (do not consider variants like RoBERTa, Longformer etc.) ?

- BERT has a limit of 512 input tokens. When query & an entire document is together feeded in, 512 might be too less a number, since documents are long
- Every (query, doc) pair processing induces a lot of latency during inference

(b) (3 points) Provide at least 3 different ways in which the BERT's limitations can be addressed during document retrieval. Valid answers should provide a clear equation / diagram (if any) of each approach.

~~→ ColBERT can be~~

→ Biencoders can be used. ~~How do doc & query and~~ → can be stored in disk



The doc emb. to be stored in disk reduced the latency greatly

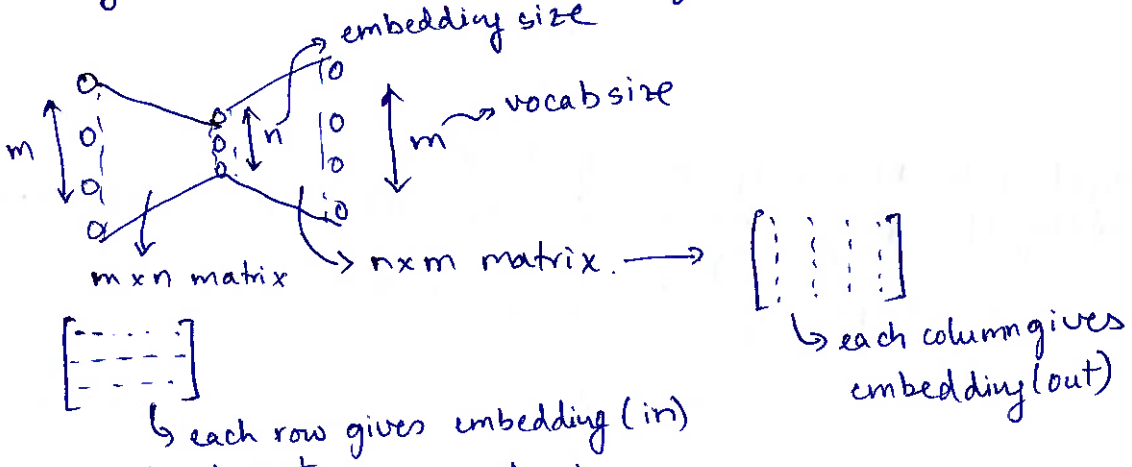
→ ~~to~~ Colbert can be used

→ Polyencoders can be used
In this way doc terms can be compared to multiple views of the query.

(Continued....)

7. (3 points) While viewing word2vec from the lens of SVD, we remarked that it effectively throws away the context matrix C, and retains only the weight matrix W. If we were not do throw away C, what can we do with it? Explain - possibly with an example.

In w2v, we only consider the in-~~context~~ weight matrix. We discard the matrix which is going away from the hidden layer. It can be used to generate a second set of embeddings per word, which can be used along with the earlier embeddings to get more semantic info.



* in-in and out-out similarities for a word is higher with other close words which carry similar meanings.
* in-out similarities are higher for words which co-occur together

these two properties would help in query expansion & then re-ranking

Eg. For a word like "machine"
in-in & out-out would capture similar words like "computer", "processor", "system" etc.

in-out similarities would capture words like "learning", "building", "intelligence" etc.

8. Consider the ColBERT model of retrieval using late-interaction paradigm over BERT base. In this context, explain the following:

(a) (2 points) What is the advantage of late-interaction paradigm proposed in ColBERT?

The documents side of preprocessing could be done offline. This greatly reduces the latency, considering that now only the query preprocessing has to be done, which will be much less expensive due to smaller size of queries

(b) ColBERT uses *MaxSim* operator to implement the late-interaction paradigm.

i. (2 points) Write the equation of *MaxSim* operator and explain each component.

The inputs to the MaxSim op. are $q, d_1, d_2, \ldots, d_n$ (say), where q is the embedding of a query term (for a given query) and d_i's are embeddings of each doc term (of a given doc)

it returns $\max_{1 \le i \le n} sim(q, d_i)$ as the similarity of the most similar doc term to the query term.

$sim(\cdot)$ could be any similarity function. Eg. cosine

ii. (2 points) What is the computational cost of each *MaxSim* operator?

For each similarity computation, it does K operations (one per embedding dimension)

n similarity to compute $\Rightarrow$ $O(nK)$ where n is the number of document terms

K is the embedding dimension

(c) (2 points) Give one-line explanation of at least two ways we studied in the course for further speeding up the ColBERT retrieval.

We can use CITADEL or also memorize the document embeddings offline.

9. (6 points) Answer the following questions in the context of differentiable search index (DSI) framework of using generative retrieval models for ranked retrieval.

   Following are the three essential features of an end-to-end ranked retrieval model.

   1. **Exclusivity.** Document content in the index should have a one-to-one correspondence with the ID.
   2. **Completeness.** Key information of the documents should be stored in the index as completely as possible to avoid loss of information related to the query. This also indicates the ability of the index to achieve as high a recall as possible for any query.
   3. **Relevance ordering.** The model should be able to output documents in the order of their relevance to the query (or estimated probability of relevance to the query).

   Which of the above essential features do you think DSI model have and under which docid design? Explain your reasoning for each feature separately.

~~to be considered for doc id design~~

Exclusivity

DSI Model might not be necessarily exclusive. so Doc ids are learnt here and so, they should be more flexible towards semantics over one-one correspondence.
The doc-id design in focus is the doc → query model. Here, a language model is used as the doc 2 query model and the document is used to generate a query.
In this context, exclusivity cannot be guaranteed.

(Continued....)

Completeness

The mentioned doc-id design captures a lot of semantics since queries are generated here, such that that doc is pointed to. The query generated will store info about the topics this doc is about, the frequent words and other such info which is vital to that particular doc.

Also, The retrieval model could be adjusted accordingly such that any one captured semantic wrt the query would lead to that doc being marked as relevant. In this way, a lot of docs close to the query would be reported relevant. The resulting recall would be high.

Relevance Ordering.

There could be a similarity score which could be reported by the retrieval model b/w the query and the generated query from doc.
of course, this might not be just a simple similarity function b/w vectors but rather a similarity score returned by a retrieval model. This could be ordered accordingly.

10. (3 points) In our discussion of PageRank, we alluded the underlying assumption of a "Random Surfer" who is simply following outgoing links from a page with uniform probability, or jumping to an arbitrary page in the graph.

    Clearly this is quite an unrealistic model. A more realistic model called "Intelligent Surfer" was proposed by Richardson and Domingos in their NeurIPS'01 paper. This surfer starts with an information need, expressed as a k-term query $q = \{q_1, q_2, ..., q_k\}$. When the surfer is at a page v_i, she chooses the next page to transition v_j based on the (relative) relevance of the contents of the page v_j to the query. When there are no links to follow from a page, either because there are no outgoing links or none relevant to the query, she takes a random jump to a page in the Web graph with a probability proportional to its relevance to the query.

    Based on the above description, write the equation for the formulation of the iterative score computation in case of intelligent surfer model.

$\frac{n-1}{n}, \frac{n-2}{n}, \ldots \frac{1}{n}$

$p^{(t)} = (\alpha M^T + (1-\alpha)\vec{e}\, p^T(c))\, p^{(t+1)}$ [- - -]

$p^{(i)}$ → the state of the random surfer at time i

α → probability with which it chooses to follow the next link from the page

$\vec{e}$ → matrix of size 1 × n where n is the number of topics

Consider the prob. transition matrix of size n × n. n is the no. of pages.

M = [-] for each row, give a score of $n-1$, $n-2$, ... and so on → (*) to each outgoing linking page where n is the no. of links from a page.

Then apply softmax to convert into probabilities.

We have the transition matrix now.

where there is a link, $p_j^{(t)} = \sum_{i=1}^{n} M_{ji}\, p_i^{(t-1)}$ where $p_j^{(t)}$ is the prob. state of j^{th} page at time t.

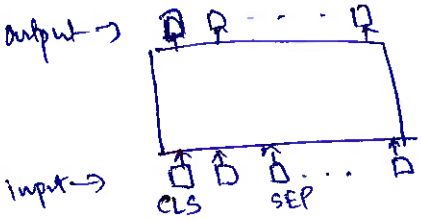
When there is no link, $p_j^{(t)} = 0$, then take $n = n$ in (*) & apply softmax. Let this value be $f(j)$ for the j^{th} page

Now transition to that page ie,

$$p_j^{(t)} = \begin{cases} \sum_{i=1}^{n} M_{ji}\, p_i^{(t-1)} & \text{if } \sum_{i=1}^{n} M_{ji}\, p_i^{(t-1)} \neq 0 \\ f(j) & \text{else} \end{cases}$$

11. (4 points) Select **any ranked retrieval model** we have covered in this course —ranging from BM25 to generative retrieval — and describe of the model in all its stages, including but not limited to: training (if any), index building (if any), inference/retrieval and its implementation for efficiency.

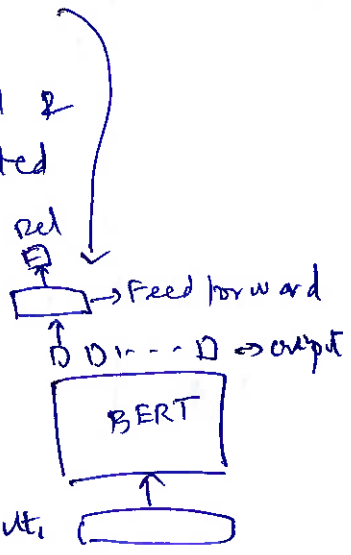
Mono BERT First tokenization model is selected.



Here, training is done on a huge corpus with consecutive sentences being inputted. The embeddings (term, position, section) are added & provided. First token is CLS, and SEP is there. For training, a random token is masked & then it is trained on the probability distribution generated on the vocab by the corresponding output token.

Fine-tuning is for IR is done as query, doc pairs and given to the model. It is trained on the CLS token which is further fed to a feed-forward network to generate the relevance value.

For inference, query, doc pairs are given to the model & relevance value is generated after CLS token is collected at the output end.

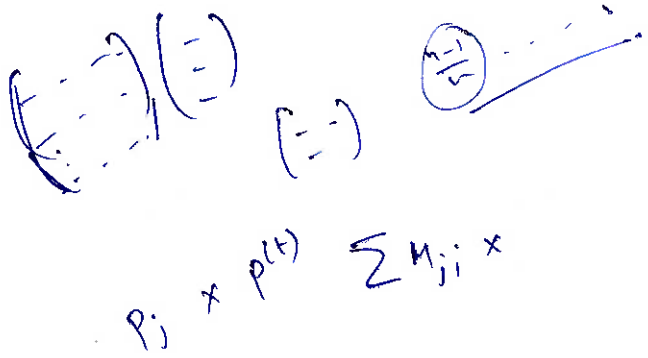


Passage is brokendown and scores are collected for better accuracy since there is a limit on the no. of input tokens

End of the paper

Rough Work - Will not be evaluated

2
6 - b
8 - c
10



$p_j \times p^{(t)} \sum M_{ji} \times$

Rough Work - Will not be evaluated