Neural Information Retrieval 2022: Exam

Project submission deadline: 14 June 2022 at 23:59
This description is likely to be adapted throughout the course.

1 Exam

The exam of this course consists of an individual project and an individual oral exam. The final course grade is computed on the basis of the project and the oral exam as a whole.

The oral exam consists of a presentation of the project by the student, followed by questions by the examiners on both the project and the contents of the whole course.

The project must be submitted by June 14, 2022 (Note that there are three parts that need to be submitted, see next section). A student who has not submitted a project will not be allowed to have an oral exam and will automatically fail the course.

The rest of this document provides information on the project.

2 Project Logistics

In this project you will develop an IR system for the news domain. **All parts of this project are compulsory**. The project should be completed **individually**. You should submit:

- A **report** (`.pdf` file) detailing the project, what you have implemented, and your results and observations;
- **Code** (`.zip` file) to run your experiments and **documentation** (`read-me` file) on how to run it;
- Your **retrieval runs** on all training and test topics (`.txt` file) in trec-eval format (details on this are given in Section 3.5.1);
- Do **NOT** include the **dataset** in your code folder.

You need to hand-in the final submission on DigitalExam: you need to submit the report as main file and the code and retrieval runs as attachment. The submission deadline is no later than **June 14 2022 at 23h59**. The format of the report should be a PDF document using the ACL template[1], no more than **6 pages** (not including references, if needed). For the code submission, you can use any programming language.

[1] `https://github.com/acl-org/acl-style-files`

Academic Code of Conduct

You are welcome to discuss the project with other students, but sharing of code or text is not permitted. Copying code or text directly from other students will be treated as plagiarism. Please refer to the University's plagiarism regulations if in doubt. For questions regarding the project, please ask on the Absalon discussion forum.

3 Project Description

The goal of the project is to implement and evaluate search algorithms as part of a search engine for helping people retrieve factual information in the news domain.

This project has the following three learning objectives:

- Designing appropriate strategies for storing and ranking information;
- Evaluate text retrieval models;
- Discuss and present experimental results.

Details are given below.

3.1 Data pre-processing and indexing

3.1.1 Collection (week 17)

You will use the NIR2022 dataset.[2] This dataset is a collection of newswire documents. For each document, you have access to several fields, such as title, or abstract, for instance.

This dataset includes queries (also known as topics) and also ground truth in the form of annotations (also known as relevance assessments) specifying which document is relevant to which query. You will be given access to 200 topics and their relevance assessments, so that you can train your methods. Later on in the project, you will be given some unseen topics, on which you need to run your IR system (see Section 3.5).

Relevance assessments are on a three point scale: highly relevant (2), relevant (1), or not relevant (0). Whenever a document is not assessed for a given topic, you can consider it as *not relevant*.

On Absalon, you can find the relevance assessments (qrels) of the training topics[3]. The relevance assessments of the unseen test topics will be used for a competition, as described in Section 3.5, and you will not have access to them.

As a first step of this assignment, we recommend that you have a look at the documents and topics. What do the documents look like? How is each document structured? Are different fields separated? How are topics structured? Looking at the topic of the query, would you expect the retrieval task to be easy or difficult?

[2] `https://absalon.instructure.com/courses/56884/files/folder/project`.
[3] `https://absalon.instructure.com/courses/56884/files/folder/project`

During this week, you should also download one of the following IR systems: Terrier,[4] Indri,[5] Elasticsearch,[6] Anserini,[7] etc.

3.1.2 Indexing (week 18)

The purpose of this part is to familiarize yourself with the dataset and create the index that will be used by your retrieval algorithms. You should complete the following tasks:

- Load the data and compute relevant statistics: number of documents, document length, number of assessed documents per topic, average topic length, etc.
- Index the dataset to optimize speed and performance in finding relevant documents for a search topic (you will use this index in the remaining parts of the assignments). Without an index, your search system would have to scan every document in the dataset, which is very time consuming. We suggest that you use existing tools and libraries (e.g., Terrier,[8] Indri,[9] Elasticsearch,[10] Anserini,[11] etc.) to build your index. Python wrappers for different libraries are also available (e.g., PyTerrier,[12] Pyserini,[13] Elasticsearch-py,[14] etc.).
- You should try different approaches to optimize your index, either on the pre-processing level (e.g., stopword removal, stemming or lemmatisation), or on the index construction level (e.g., impact ordering, index pruning) and see the impact of each approach. This will result in different indices.
- After building each of your indices, you should extract information about it, such as number of documents indexed, number of unique terms, total number of terms, index size, etc. Note that using different optimization approaches results in different index statistics.
- You should include efficiency statistics (size of each version of your indices, how much time it took to build each version of your index and how much time it takes to process a query with each version of your index) in your report.

In your report, you need to include all the important details regarding the implementation of your indices, including any preprocessing of the data. Moreover, the goal of this section is to discuss and compare different indexing approaches. What are the advantages and disadvantages of your indexing approaches? What are the most and least efficient indices?

[4] http://terrier.org/
[5] http://www.lemurproject.org/indri.php
[6] https://www.elastic.co/
[7] https://github.com/castorini/anserini
[8] http://terrier.org/
[9] http://www.lemurproject.org/indri.php
[10] https://www.elastic.co/
[11] https://github.com/castorini/anserini
[12] https://pyterrier.readthedocs.io/en/latest/
[13] https://github.com/castorini/pyserini
[14] https://github.com/elastic/elasticsearch-py

3.2 Ranking Models (week 19)

3.2.1 Probabilistic Models

Once you have the index, you can start implementing your ranking models. You should do the following:

- (week 19) Tune and run BM25. You are allowed to use existing search libraries. The same library that you have used to build your index will likely implement a version of BM25.
- (week 19) Tune and run a Language Model. Differently from BM25, the basic language modeling approach builds a probabilistic language model from each document, and ranks documents based on the probability of the model generating the query [1].

In order to tune your ranking models, you should split your training queries into several folds and use cross-validation to tune your models. For example, if you split the training topics into 3 folds, you train your model using two folds and test on the remaining fold (test fold), and you then repeat by alternating the folds until all folds have been used once as a test fold. You can compute the average performance of all the test folds in order to find the best configuration. This is the final configuration of your model (tuned model) that you should use on the unseen topics that we will release later on.

The above is a description of the minimum that we expect you to implement, but you are allowed and encouraged to expand upon this to explore more complex ideas.

In your report, you need to include all the important details regarding the implementation of your models, as for example the library you used and how you tuned the parameters of you model. The goal of this section is to discuss and compare different retrieval approaches. You can combine each approach with the different indexing strategies you implement in Section 3.1.2. Which combination of index and ranking model is the most effective? Which one is the least effective? What affects effectiveness more, the index or the ranking model? Why do you think that some indices or retrieval models are more effective than others? Are there some features in this collection that make the retrieval of relevant documents easy or difficult? Are there differences among topics? Are there easier or more difficult topics? Why yes or why not?

3.3 Evaluation part I (week 20)

For the evaluation of your retrieval methods, you should follow the "Cranfield" paradigm which is used in the Text Retrieval Conference (TREC) and related search engine competitions. We require that you use `trec-eval`[15] for evaluation. You need to report the Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) at cut-offs 5, 10, 20. You are not limited to those measures; you can use further evaluation measures and explore different cut-offs. You are also expected to report the mean response time of your methods.

Different models will perform differently also depending on the evaluation measure. You have to describe your models and how they are tuned/trained,

[15] `https://github.com/usnistgov/trec_eval`

as well as the evaluation results. It is up to you to decide whether to report the results in tables, plots, box-plots, etc. Remember to give insights on which models work the best and why, highlighting your design decisions.

3.4 Neural Information Retrieval methods (weeks 21-22)

Re-ranking It is required that you implement the following advanced approaches:

- Word embeddings can be used to extend the query itself, without modifying the ranking function. First, use word embeddings to do query expansion as done by Kuzi et al.[16][4]. A popular word embedding choice is the word2vec pre-trained on Google News corpus,[17] but you can decide on another embedding. Second, expand on this by using contextualized word embeddings, such as BERT [3]. You can use BM25 or any other model from Section 3.2.1 as ranking function. Third, investigate manual term expansion techniques [6]. How does applying different weights to expanded terms affect retrieval performance?

- A different approach you should try is to use BM25 or any other model from Section 3.2.1 to generate an initial ranking, and then re-rank the top K[18] documents using contextualized embedding approaches such as BERT, or Sentence-BERT [5]. To do so, you have to build the query and document representations using these approaches and then compute the similarity score between them. The similarity score between each document and query can be used to re-rank the documents. You can also get some inspiration from [2]. Does training these models on your data lead to better retrieval?

If the authors of the above publications have trained a new method or model and they have released the trained model, you do not need to train it too; you can simply use it and cite them (your citation should include the url where the released model can be found).

The above is a description of the minimum that we expect you to implement, but you are allowed to expand upon this to explore more complex ideas.

In your report, you need to include all the important details regarding the implementation of your models, as for example the library you used, how you tuned the parameters of your models and any external resource you used. The goal of this section is to discuss and compare your models with those implemented in Section 3.2.1. Is your advanced model performing better or worse than the simpler models in Section 3.2.1? Why or why not? Are your results aligned with those reported in the literature? How can you improve your models? Are there some features in this collection that make the retrieval of relevant documents easy or difficult? Are there differences among topics? Are there easier or more difficult topics? Are the easy and difficult topics the same you found in Section 3.2.1?

[16] `http://publish.illinois.edu/saar-kuzi/files/2017/10/w2v_cikm16.pdf`
[17] `https://github.com/mmihaltz/word2vec-GoogleNews-vectors`
[18] Typically, $K \in \{100, 500, 1000\}$.

3.5 Evaluation II - unseen queries (week 23)

To test the effectiveness of your models in a more realistic setting, we will evaluate your models on a test set of unseen topics. You are required to take part in this evaluation and to discuss in your report your training results compared to test results. For this, you need to discuss the difference in terms of effectiveness scores between the training topics and the test topics. Are your models performing comparably on the training and test topics? Is the measure score higher or lower on the test topics than the training topics? Why?

We will release the unseen topics on Absalon by Monday 6 June at 09h00. We ask you to submit a minimum of five runs, in trec-eval format, each of them in a separate .txt file, and all of them uploaded as one .zip file on Absalon by **Wednesday 8 June at 16h00** at the latest. We ask you to choose a name for each run. The name should be the filename of the .txt file that contains that run. Each run should have a different name. Run names should be in the following format: {any 3 small letters of your choice}{any three numbers between 0-9 of your choice}{any three capital letters of your choice}{any three numbers between 0-9 of your choice}. For example, the following are valid run names:

- dog357IJS302
- lod547ASE667
- qok312POW987
- aqw769GHH095

The following are examples of invalid run names:

- dg357IJS302
- lod547.ASE
- 312POW987qok
- aqw7650GHH095

We will then evaluate all the runs we receive and we will release the results on Absalon by Friday 10 June at 16h00 at the latest. You should make sure that you remember the name of your own runs, so that you can look at their performance and include a discussion of this in your report. The rest of this section provides more details on this.

3.5.1 Format

The submission format of your five runs will follow the standard TREC run format. The submission format of a ranked result list (run) is as follows:

```
qid Q0 docno rank score tag
```

The fields should be separated with a white space. The width of the columns in the format is not important, but it is important to include all columns and have some amount of white space between the columns. The above fields are:

- `qid`: the topic number;
- `Q0`: unused and should always be Q0;
- `docno`: the official document id number returned by your system for the topic `qid`.
- `rank`: the rank where the document is retrieved;
- `score`: the score (integer or floating point) that generated the ranking. The score must be in descending (non-increasing) order. The score is important to handle tied scores (`trec_eval` sorts documents by their scores values and not their ranks values);
- `tag`: your run name, in the format described above. **Each run should have a different tag**.

An example of a run is shown below:

```
1 Q0 doc1 1 14.8928003311 dgp357IJS302
1 Q0 doc2 2 14.7590999603 dgp357IJS302
1 Q0 doc3 3 14.5707998276 dgp357IJS302
1 Q0 doc4 4 14.5642995834 dgp357IJS302
1 Q0 doc5 5 14.3723001481 dgp357IJS302
...
```

The submitted runs should contain at most the top 1,000 documents for each test topic, with all the topics in each file. The submission file must be in `.zip`. The topics for the test set will be released on Absalon.

You will be asked to submit your runs by **8 June 2022 at 16:00**.

3.5.2 Evaluation of Runs

We will evaluate the submitted runs with the following measures:

- Mean Average Precision (MAP);
- Normalised Cumulative Gain (nDCG) at cut-offs 5, 10, 20;

We will post the results of the evaluation on Absalon by 10 June at 16h00. If a run is not in the correct format and cannot be processed by trec-eval, we will ignore it.

Recall that we will release the qrels only for the training topics, thus, you will not have the qrels for the test topics. Therefore, you have to set your own evaluation to train and tune your methods on the training topics.

4 Requirements of the written report that describes your project

The following considerations are applicable to all sections of this project. You need to keep them in mind when you write your report.

In all cases, you should describe *what* you tried, what worked, what did not work, and *why* you think it did or did not work. For example, did you use an off-the-shelf method? How did you adapt it for the given task? Why did this adaptation work or not? Did you do some preprocessing or cleaning of the dataset? Was is it useful? Why or why not?

It is not enough to report the evaluation scores of your methods. You need to explain the reasons why you think we see these scores.

You need to show in the report that you have tried to understand why you got these specific evaluation scores, regardless of whether they are high or low.

You should also describe the limitations of what you did. What could have led to improvements in model performance? How could you have approached this task differently? What was particularly challenging about working with this dataset and why do you think that was? This discussion does not require further experiments, but requires you to examine your experimental results, critically think about your choices and the assumption you made, hypothesise on how you can overcome some limitations and improve your solution. Your discussion should be based on evidence, such as lecture material and relevant literature.

Finally, in all cases, you should cite relevant literature which informed your choices in terms of modeling, analysis, etc.

References

[1] S. F. Chen and J. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. *Comput. Speech Lang.*, 13(4):359–394, Oct. 1999. `https://dash.harvard.edu/bitstream/handle/1/25104739/tr-10-98.pdf?sequence=1`.

[2] Z. Dai and J. Callan. Context-aware Sentence/Passage Term Importance Estimation for First Stage Retrieval. *arXiv preprint arXiv:1910.10687*, 2019. `https://arxiv.org/abs/1910.10687`.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] S. Kuzi, A. Shtok, and O. Kurland. Query Expansion Using Word Embeddings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 1929–1932, New York, NY, USA, 2016. Association for Computing Machinery. `http://publish.illinois.edu/saar-kuzi/files/2017/10/w2v_cikm16.pdf`.

[5] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[6] T. Schoegje, C. Kamphuis, K. Dercksen, D. Hiemstra, T. Pieters, and A. de Vries. Exploring task-based query expansion at the trec-covid track, 2020.