Information Retrieval Project (IE 691)

Prof. Dr. Goran Glavaš
Fabian David Schmidt
Robert Litschko

Data and Web Science Group
Fakultät für Wirtschaftsinformatik und Wirtschaftsmathematik
Universität Mannheim

CreativeCommons Attribution-NonCommercial-ShareAlike 4.0 International

Team project

- Tentative schedule
  - Topics published: March 15
  - Team formed & topics selected: Mar 22

  - **Project coaching:**
    - Two (**optional**) sessions: April 19, May 31
    - We check the progress of your projects
    - Help you resolve dilemas and problems you might be facing

  - **Project presentations:** June 14
    - Present what you did:
      - methods/models, implementation, evaluation
    - 10-15 minutes per team
    - **All team members should present and clearly state what their contribution was**
      - We will ask questions to all team members

Team Project

- **Purpose:** „hands-on” experience implementing and evaluating information retrieval model(s) and performing IR tasks
  - Best way to understand something is to (try to) implement it

- Other goals:
  - Experiencing teamwork
  - Learning how to coherently and concisely report on the results of the work
  - Exercising how to clearly present results of your work

Topics

1. Learning to Rank (Supervised Retrieval; Neural Rankers)
2. Question Answering from Wikipedia articles
3. Cross-Lingual Sentence Retrieval
4. Efficient Vector Space Retrieval

- You can propose your own topic as well
  - Has to undoubtedly be a retrieval topic (or closely related: question answering; summarization, etc.) and of similar scope/difficulty as the four above

Topic 1: Learning to Rank (Supervised Retrieval)

- In some settings, we have enough relevance judgements to train supervised retrieval models
- **Learning to rank** (L2R, LETOR): training supervised machine models for IR
- **Task:**
  - Implement and evaluate two L2R models
    - One *point-wise* L2R model
    - One *pair-wise* L2R model
  - Design good, informative features for both models
    - Different unsupervised ranking functions can be used as features
  - Evaluate the performance of the models on test collections

Topic 1: Learning to Rank (Supervised Retrieval)

- Point-wise L2R model
  - One training instance is a query-document pair (q, d)
  - You are predicting whether the document is relevant for the query
  - Ranking: order documents by the classifier's confidence
- Pair-wise L2R model
  - One training instance is a triple (q, d1, d2) consisting of a query and two documents
  - You are predicting which of the two documents (first or second) is more relevant for the query
  - Ranking: merging pairwise decisions into consistent ordering
- Dataset:
  - TREC DL 2019 dataset for passage Re-ranking: https://microsoft.github.io/msmarco/TREC-Deep-Learning-2019#document-ranking-dataset
  - Paper: https://arxiv.org/pdf/2003.07820.pdf

Topic 1: Learning to Rank (Supervised Retrieval)

- **Full task** (for max. 100 points):
  - **Own** implementation of both L2R models (pairwise and pointwise)
  - **Own** implementation of feature computation
  - **Own** implementation of evaluation

- **Reduced task** (for max. 70 points):
  - **Own** implementations of features, but you may use
  - Existing implementations of L2R algorithms
  - Existing implementations of evaluation metrics (MAP, MRR, NDCG)
  - RankLib – a L2R library you may use
    - https://sourceforge.net/p/lemur/wiki/RankLib

Topic 2: Trivia Question Answering

- **Task:** Reading comprehension / question answering
- Given a document collection and a question:
  - 1) Retrieve the document containing the answer to the question
  - 2) Identify / extract the answer

- **Dataset**:
  - TriviaQA: 100K+ training triplets <question, document, answer>
  - Documents are Wikipedia articles, answers are named entities from the articles
  - Paper: *Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension*.
  - Dataset: http://nlp.cs.washington.edu/triviaqa/

Topic 2: Trivia Question Answering

- **Full task** (for max. 100 points):
  - **Own** implementation a document retrieval model
    - Traditional vs. neural document retrieval model (with questions as queries)
  - **Own** implementation of an answer extraction model
    - Some near-SOTA neural answer/span extraction model is expected to be implemented
  - **Evaluation of document retrieval performance**
  - **Evaluation of answer extraction performance**

- **Reduced task** (for max. 70 points):
  - Own implementation of a (one, any) document retrieval model
  - May use existing implementations answer extraction modules

Topic 3: Cross-Lingual Information Retrieval (CLIR)

- **Cross-lingual retrieval**: query is in a different language from document collection
- Creating a retrieval system, that can, given a sentence in one language recognize its translation from a large collection of sentences in another language
- **Task**:
  - Implement a **supervised classifier** for cross-lingual retrieval (L2R)
  - Implement an **unsupervised measure** capturing cross-lingual semantic similarity between queries and documents
  - Both the supervised classifier and the unsupervised measure can be based on **multilingual / cross-lingual word embeddings** or **multilingual text encoders**
  - **IR evaluation** of both the supervised and unsupervised scorers

Topic 3: Cross-Lingual Sentence Retrieval

- Inducing a multilingual embedding space from monolingual word embeddings
  - Many ways to do it:
    - https://arxiv.org/pdf/1902.00508.pdf
    - https://arxiv.org/pdf/1710.04087.pdf
    - https://www.aclweb.org/anthology/2020.acl-main.675.pdf

- Number of pretrained multilingual text encoders:
  - Comparative evaluation of multilingual encoders for sentence-level and document-level CLIR:
  - https://arxiv.org/pdf/2101.08370.pdf

Topic 3: Cross-Lingual Sentence Retrieval

- **Full task** (for max. 100 points):
  - **Own** induction of multilingual word embedding space
  - Implementation of an **unsupervised ranking function** for recognizing sentence translation based on (a) cross-lingual word embeddings and (b) multilingual sentence encoders
  - Implementation of a **supervised classification model** for recognizing sentence translation based on (a) cross-lingual word embeddings and (b) multilingual sentence encoders
  - Evaluation of the models on EuroParl datasets
- **Reduced task** (for max. 70 points):
  - You may use some pre-trained multilingual word embeddings
    - E.g., https://github.com/Babylonpartners/fastText_multilingual
      https://github.com/facebookresearch/MUSE
  - You may implement and evaluate only the supervised prediction of sentence translation pairs
- **Datasets**:
  - EuroParl parallel corpora: http://opus.nlpl.eu/Europarl.php (sentence-level CLIR)
  - WikiCLIR DE-EN Wikipedia dataset: https://tinyurl.com/wikiclir (document-level CLIR)

Topic 4: Efficient Vector Space Retrieval

- **Efficient IR system** needs to be able to retrieve results, in real-time, from very large document collections
- The goal is to implement the **Vector Space Model** model with all „tricks" for efficient retrieval
- Task:
  - Implement a **tiered index**, with the configurable number of tiers
  - Implement **pre-clustering** of documents in the collection using randomly chosen *leaders* (i.e., vectors in the same space as document vectors)
  - Transformation of TF-IDF VSM document vectors into the lower-dimensional vector space via *random projections*
  - Evaluate different variants in terms of (trade-off of)
    - **(1) retrieval performance (e.g., MAP),**
    - **(2) retrieval speed**

Topic 4: Efficient Vector Space Retrieval

- **Full task** (for max. 100 points):
  - **Own** implementation of the basic VSM model (TF-IDF weighting + cosine ranking)
  - Own Implementation of all speed-ups (tiered index, pre-clustering, random proj.)
  - Evaluation of all VSM variants in terms of both retrieval performance and efficiency
- **Reduced task** (for max. 70 points):
  - Implement the regular inverted index instead of the tiered index
  - You don't have to implement random projections
  - In other words, standard VSM with regular inverted index + document pre-clustering
- **Datasets**:
  - TREC DL Document Ranking Dataset: https://microsoft.github.io/msmarco/TREC-Deep-Learning-2019#document-ranking-dataset
- **Tip**:
  - FAISS: a library for fast computation of vector similarity/distance
  - https://github.com/facebookresearch/faiss

Organization

- Form groups of **3 students**
  - All students must contribute to the project
    - Ideally, equally (We will ask about individual contributions)

- Each group is allowed to pick a topic **they like the most**
  - Theoretically, all groups **could** pick the **same topic**
    - More competition, we will directly compare your results when evaluating

Organization

- Topic selection and team forming
  - **Deadline**: Monday, **March 22** (23:59)
  - **Send the email to Robert & Fabian with:**
    - Student names and IDs (Matrikelnummer)
    - Selected topic
  - If you cannot find a team on your own, let us know – we will form teams from students without a team
    - Do this by this Friday, March 19

- Emails to: litschko@informatik.uni-mannheim.de
  fabian@informatik.uni-mannheim.de

Participation

- **Important**:
  - We have **students** on the **waiting list** for the module
  - If you're thinking about **dropping this course**, now would be the **last time** to do it
    - If so, please inform us as soon as possible
  - Dropping at a later point is **very inconvenient**:
    - (1) not fair to your colleagues in the team, who will have to do your part of the work too
    - (2) for motivated students on the waiting list, whose spot in the course you'll have taken and wasted

Submission & Evaluation

- **Submitting** the project results is via **ILIAS**
  - Upload results on ILIAS
  - Code (software) as one archive and project report as one PDF file
  - **Deadline** for submission: **June 13 (Sunday), 23.59**

- **Evaluation**
  - **Implementation** **(60%)**, **report** **(30%)**, and **presentation** **(10%)**
  - We will assign a score to the group
    - Multiplied by the numbers of members in the group
    - We will ask the group members to distribute the total sum of points among themselves
    - This distribution of points **might** be taken into account for individual points
  - We will also estimate **individual contribution** to the final project results

Submission & Evaluation

- Points (max. 100) assigned to the group
  - Group members propose the distribution the points among themselves
  - Example: we assign **72** points to a group of **3** students, students then propose how to distribute **3*72 = 216** points among themselves
    - A single student **cannot** be assigned more than maximal **100** points
  - **All students should contribute** – **we will check!**
  - Our final decision on points **may differ** from the distribution proposed by the group