Information Retrieval – Assessment Test

- date: **December 10, 2024 (Tuesday); start time: 17:10** (CET (Polish) time (we start writing at 17:10, so be there earlier; no other date will be possible; if you are late, the time will not be extended);
- place: **Room L051 (ground floor in the BT building)**; please check earlier where this room is located;
- **duration: 100 minutes** (17:10 – 18:50) – you need to be very well prepared; the number of tasks will be high, but the designated time is fully sufficient for solving them (provided that you are fluent in solving them);
- everyone will get her/his individual assignments (the tasks will be the same for subsets of students, but the data and numbers will differ from one student to another) – you will get around 16 tasks (4 pages filled with text and empty spaces for providing your solutions); you will get at least 1 task from each of the 7 lectures;
- **you are allowed to use lecture materials, your own notes, and a simple calculator**; the materials and notes can be printed or digital; you are allowed to use your own laptops; however, it is strictly prohibited to use the Internet (Internet connection needs to be switched off before you enter the exam room); also, you cannot use any programming code for solving the tasks; once this is noted, this means the exam's end for such a person; you will need a simple calculator to perform some calculations (e.g., multiplying three numbers or computing a logarithm), so be prepared and don't lose time during the test; **any attempt of making photos of the test (no matter solved or not) with any device will result in its immediate failure;**
- **all forms of communication between students are strictly prohibited during the exam**; once this is noted, you get the first warning (-20% of points); the second warning means you fail the entire exam immediately;
- the number of points for each task will be specified in the squared brackets (e.g., [2p], [3p]);
- we will apply standard scheme for derving the marks: <50% - 2 and failed; [50%, 60%) – 3.0; [60%, 70%) – 3.5; [70%, 80%) – 4.0; [80%, 90%) – 4.5; [90%, 100%] - 5.0;
- write down your first and last names at the first page immediately after seeing the exam;
- the difficulty of the test will be similar to the difficulty of the exercises that you were given after each lecture, but surely the tasks will not be the same as in the assignments; nevertheless, if you haven't solved and understood them, the chances you will pass are close to zero;
- **the 2nd term will be organized on January 7th (Tuesday), 2025 (for those who would fail the first term);** the date and the time will be confirmed with the results of the first term; no other dates are possible as I am out of the university and Poland for 4 weeks starting in the 2nd half of January.

Types of tasks that you may expect at the assessment test

These are only general formulations of what you may expect (if the exact text of the task during the exam doesn't mention something, don't lose time, and don't do it). Below, you can find 23 types of tasks – they will appear all in different sets of tasks for various students, but each of you will get around 16 of these tasks. This list is to avoid any surprises during the test so that you don't lose time thinking about how the exam will look like or whether some aspect will be present or not.

1. **[Inverted Index]** Given documents in a raw form (lists of terms), construct a record-based (block) inverted index or a positional (term-based) inverted index with document and term frequencies, using a linked list representation. Answer Boolean and/or phrase queries using the index.
2. **[Query optimization]** Given the posings list sizes, recommend an optimal query processing order for Boolean queries.

3. **[Log analysis]** Given the entries in the log file and, possibly, a service structure, identify users and sessions using H1, H2, or HREF.
4. **[Navigational patterns]** Given the log file in the form of sessions with visited pages, draw the Markov chain and use it to compute designated probabilities.
5. **[Similarity]** Given documents in a raw form (terms), compute the Jaccard similarity measure for some pairs.
6. **[TF, TF-IDF, cosine]** Given a set of terms, a collection 2-4 documents, and a query, show their TF (with normalization by max) and TF-IDF representations, compute the lengths in TF-IDF representation and the cosine similarities for all documents and the query.
7. **[LSI]** Given the original term-document matrix and the results of SVD, indicate the importance of concepts, the mapping between the concepts and documents or original terms. Compute matrix C_k based on k (1-2) most important concepts. Compute the Frobenius norm value.
8. **[Evaluation measures for unranked sets]** Given the set of relevant and retrieved documents, compute precision, recall, and F.
9. **[Evaluation measures for ranked lists]** Given a ranking of returned documents (R N) compute precision at k, recall at k, MAP, and R-precision.
10. **[PageRank]** Given the web's structure, show the stochastic matrix M, write down the equations for PageRanks with or without a damping factor, justify which page has the greatest/least PageRank. The same for the Inverse PageRank or TrustRank.
11. **[HITS]** Given the web's structure, show the adjacency matrix L, write down the equation for hubs or authorities in the function of authorities or hubs, respectively. Given the principal eigenvectors of LL^T or L^TL, indicate which page has the greatest/least hub or authority score, and justify why it makes in view of the other hub/authority scores.
12. **[Rocchio Relevance Feedback]** Given the original query, sets of (1-2) relevant and non-relevant documents, show the revised query obtained with the Rocchio relevance feedback. Use a binary or bag-of-words representation.
13. **[Levenshtein]** Compute the Levenshtein distance for two terms. Fill the entire matrix using dynamic programming and provide the final answer.
14. **[Collaborative filtering]** Given the rating matrix and relevant similarity measures, employ user- or item-based collaborative filtering for a specific value of K with the average, weighted average, or the weighted modification of nearest neighbors' averages to predict the unknown rating.
15. **[Balance]** Use a simplified Balance algorithm to select the ads for the query stream. Compute the competitive ratio.
16. **[Adwords]** Use a simple Google Adwords algorithm to rank the advertisers. Indicate whose ad would be selected.
17. **[Suffix tree]** Show a complete suffix tree for a string. Answer some additional questions based on the tree. No need to show all the stages; just the final tree.
18. **[Suffix array]** Build a suffix array using the qsufsort algorithm. Show all arrays (A[i], V[A[i]], V[A[i+h]]) in each stage.
19. **[Zipf]** Use Zipf's law to compute the rank of a term based on the number of its occurrences, a comprehensive number of all tokens, and a parameter A. How many terms have at least this number of occurren-

ces? Knowing the number of occurrences of the 1st, 2nd, or 3rd most common term, predict the number of occurrences for the 1st, 2nd, 3rd, …, most frequent term.

20. **[Heaps]** Use the Heaps' law to predict the number of different terms for a certain number of tokens and values of parameters k and b.
21. **[Delta/gamma coding]** Encode some number in gamma or delta.
22. **[Delta/gamma decoding]** Decode some number in gamma or delta.
23. **[MapReduce]** What are the roles of mappers, reducers, combiners, and partitioners? What is the order of different phases in the MapReduce framework? Whose output is whose input? What is a job? Who is called mini-reducer? What is in-mapper combining about? What is the role of identity mapper? Can reducers be used as combiners? Is combiner always guaranteed to be executed? For which Information Retrieval tasks it pays off to use MapReduce? Etc.