1.37 Information Retrieval

- **Course name:** Information Retrieval
- **Course number:** XYZ
- **Subject area:** Data Science

1.37.1 What subject area does your course (discipline) belong to?

Computer systems organization; Information systems; Real-time systems; Information retrieval;World Wide Web

1.37.1.1 Key concepts of the class

- Data indexing
- Relevance and ranking

1.37.1.2 What is the purpose of this course?

The course is designed to prepare students to understand background theories of information retrieval systems and introduce different information retrieval systems. The course will focus on the evaluation and analysis of such systems as well as how they are implemented. Throughout the course, students will be involved in discussions, readings and assignments to experience real world systems. The technologies and algorithms covered in class include machine learning, data mining, natural language processing and so on.

1.37.1.3 Course Objectives Based on Bloom's Taxonomy

- What should a student remember at the end of the course?

- Terms and definitions used in area of information retrieval,
- Search engine and recommender system essential parts,
- Quality metrics of information retrieval systems,
- Contemporary approaches to semantic data analysis,
- Indexing strategies.

- What should a student be able to understand at the end of the course?

- Understand background theories behind information retrieval systems,
- How to design a recommender system from scratch,

- How to evaluate quality of a particular information retrieval system,
- Core ideas and system implementation and maintenance,
- How to identify and fix information retrieval system problems.

- What should a student be able to apply at the end of the course?

- Build a recommender service from scratch,
- Implement proper index for an unstructured dataset,
- Plan quality measures for a new recommender service,
- Run initial data analysis and problem evaluation for a business task, related to information retrieval.

1.37.1.4 Course evaluation

Table 1.101: Course grade breakdown

Type	Default points	Proposed points
Labs/seminar classes	50	0
Interim performance assessment	25	70
Exams	25	30

Labs are followed by the home works. Home works are covering 70% of the grade. 30% of the grade fall to exam session, which will be in the form of problem solving.

1.37.1.5 Exam and retake planning

Exam

Exam is conducted in a form of a model problem solving. Students are given a dataset and a relevance markup for a set of test queries. Students create a web service with a predefined interface. This service is tested in automated manner to pass predefined quality threshold. Also, the service is tested for implementation of features, stated at the exam. Those can be:

- spellchecking
- text preprocessing
- metric implementation
- heterogeneous content support (pdf, media, ...)
- index update
- ...

Retake 1

First retake is conducted in a form of project defense. Student is given a week to prepare. Student takes any technical paper from Information Retrieval Journal (https://www.springer.com/journal/10 for **the last 3 years** and approves it until the next day with a professor to avoid collisions and misunderstanding. Student implements the paper in a search engine (this can be a technique, metric, ...). At the retake day student presents a paper. Presentation is followed by QA session. After QA session student presents implementation of the paper. Grading criteria as follows:

- 30% – paper presentation is clear, discussion of results is full.
- 30% – search engine implementation is correct and clear. Well-structured and dedicated to a separate service.
- 30% – paper implementation is correct.

Retake 2

Second retake is conducted in front of the committee. Four (4) questions are randomly selected for a student: two (2) theoretical from "Test questions for final assessment in this section" and two (2) practical from "Typical questions for ongoing performance evaluation". Each question costs 25% of the grade. Student is given 15 minutes to prepare for theoretical questions. Then (s)he answers in front of the committee. After this student if given additional 40 minutes to solve practical questions.

1.37.1.6 Grades range

Table 1.102: Course grading range

Grade	Default range	Proposed range
A. Excellent	85-100	80-100
B. Good	75-84	59-79
C. Satisfactory	60-75	40-59
D. Poor	0-59	0-39

1.37.1.7 Resources and reference material

Textbook:

- Manning, Raghavan, Schütze, An Introduction to Information Retrieval, 2008, Cambridge University Press

Reference material:

- Baeza-Yates, Ribeiro-Neto, Modern Information Retrieval, 2011, Addison-Wesley
- Buttcher, Clarke, Cormack, Information Retrieval: Implementing and Evaluating Search Engines, 2010, MIT Press

1.37.2 Course Sections

The main sections of the course and approximate hour distribution between them is as follows:

Table 1.103: Course Sections

Section Number	Section Title	Lectures (hours)	Seminars (labs)	Self-study	Knowledge evaluation
1	Information retrieval basics	10	10	20	0
2	Text processing and indexing	10	10	20	0
3	Vector model and vector indexing	12	12	12	0
4	Advanced topics. Media processing	12	12	12	0
Final examination					4

1.37.2.1 Section 1

Section title: Information retrieval basics

Topics covered in this section:

- Introduction to IR.
- Boolean Model.
- Crawling.
- PageRank.
- Quality assessment.

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	1
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	0
Essays	0
Oral polls	0
Discussions	0

Typical questions for ongoing performance evaluation within this section

1. Enumerate limitations for web crawling.
2. Propose a strategy for A/B testing.
3. Propose recommender quality metric.
4. Implement DCG metric.
5. Discuss relevance metric.
6. Crawl website with respect to robots.txt.

Typical questions for seminar classes (labs) within this section

1. What is typical IR system architecture?
2. Show how to parse a dynamic web page.
3. Provide a framework to accept/reject A/B testing results.
4. Compute DCG for an example query for random search engine.
5. Implement a metric for a recommender system.
6. Implement pFound.

Test questions for final assessment in this section

1. Implement text crawler for a news site.
2. What is SBS (side-by-side) and how is it used in search engines?
3. Compare pFound with CTR and with DCG.
4. Explain how A/B testing works.
5. Describe PageRank algorithm.

1.37.2.2 Section 2

Section title: Text processing and indexing

Topics covered in this section:

- Building inverted index for text documents. Boolean retrieval model.
- Language, tokenization, stemming, searching, scoring.
- Spellchecking.
- Language model. Topic model.
- Storing index on disk.

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	1
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	0
Essays	0
Oral polls	0
Discussions	0

Typical questions for ongoing performance evaluation within this section

1. Build inverted index for a text.
2. Tokenize a text.
3. Implement simple spellchecker.
4. Build a persistent index.

Typical questions for seminar classes (labs) within this section

1. Build inverted index for a set of web pages.
2. build a distribution of stems/lexemes for a text.
3. Choose and implement persistent index for a given text collection.
4. How to compress index on disk?

Test questions for final assessment in this section

1. Explain how (and why) KD-trees work.
2. What are weak places of inverted index?
3. Compare different text vectorization approaches.
4. Compare tolerant retrieval to spellchecking.
5. How to organize compact text index on disk?

1.37.2.3 Section 3

Section title: Vector model and vector indexing

Topics covered in this section:

- Vector model
- Machine learning for vector embedding

- Vector-based index structures

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	1
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	0
Essays	0
Oral polls	0
Discussions	0

Typical questions for ongoing performance evaluation within this section

1. Embed the text with an ML model.
2. Build term-document matrix.
3. Build semantic index for a dataset using Annoy.
4. Build kd-tree index for a given dataset.
5. Why kd-trees work badly in 100-dimensional environment?
6. What is the difference between metric space and vector space?

Typical questions for seminar classes (labs) within this section

1. Choose and implement persistent index for a given text collection.
2. Visualize a dataset for text classification.
3. Build (H)NSW index for a dataset.
4. Compare HNSW to Annoy index.
5. What are metric space index structures you know?

Test questions for final assessment in this section

1. Compare inverted index to HNSW in terms of speed, memory consumption?
2. Choose the best index for a given dataset.
3. Implement range search in KD-tree.

1.37.2.4 Section 4

Section title: Advanced topics. Media processing

Topics covered in this section:

- Image and video processing
- Image understanding
- Video understanding
- Audio processing
- Speech-to-text
- Relevance feedback

What forms of evaluation were used to test students' performance in this section?

Form	Yes/No
Development of individual parts of software product code	1
Homework and group projects	1
Midterm evaluation	0
Testing (written or computer based)	0
Reports	0
Essays	0
Oral polls	0
Discussions	0

Typical questions for ongoing performance evaluation within this section

1. Extract semantic information from images.
2. Build an image hash.
3. Build a spectral representation of a song.
4. Whats is relevance feedback?

Typical questions for seminar classes (labs) within this section

1. Build a "search by color" feature.
2. Extract scenes from video.
3. Write a voice-controlled search.
4. Semantic search within unlabelled image dataset.

Test questions for final assessment in this section

1. What are the approaches to image understanding?
2. How to cluster a video into scenes and shots?
3. How speech-to-text technology works?
4. How to build audio fingerprints?