

732A81/732A92/TDDE16 Text Mining (2022)

# Information Retrieval

Marco Kuhlmann

Department of Computer and Information Science



This work is licensed under a  
[Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

# This session

- Questions and answers (15 minutes)
- Exercises (15 minutes)
- Introduction to the lab (15 minutes)

Questions and answers

# Overview of information retrieval

1. Introduction to information retrieval
2. Index construction
3. Ranked retrieval
4. The vector space model
5. Evaluation of information retrieval systems

# Project structure

- |                           |                    |
|---------------------------|--------------------|
| 1. Identify your problem  | 8 hours (w44–w48)  |
| 2. Design your approach   | 32 hours (w49–w50) |
| 3. Evaluate your approach | 32 hours (w51–w01) |
| 4. Produce your report    | 16 hours (w02)     |

# Phase 1: Identify your problem

- A good way to identify a problem is to start with a data set that you find interesting. Have a look at sites such as Kaggle!
- You could also have a look at the project abstracts from previous years. It's fine to do a project that is inspired by an old project!
- In previous years, several students have used their Text Mining project to test an idea for a thesis project.

# Project Example: Keyword Extraction from Swedish Job Ads

Employers are increasingly using information technology, such as Applicant Tracking Systems (ATS) to aid in recruiting (Laumer, Maier and Eckhardt 2015). Meanwhile, patents are being filed for automated filtering of job applications, based on keywords (Shah 2020, Dey 2012). Increasingly job applicants will need to not just list their relevant experiences, they need to do so using the correct words (Zahn 2018, Novak 2017). This report tries to develop a suitable keyword extractor to aid job applicants write their applications. For the keyword extraction two main methods are tested: term frequency–inverse document frequency (tf-idf) and a manually defined set of rules based on Part-of-Speech (POS) tagging and dependency parsing of sentences from job applications. The tf-idf approach resulted in the best recall (recall = 0.7341) of the two methods but had a low precision (precision = 0.2768). The rules-based approach was able to perform best overall (recall = 0.5209, precision = 0.5080, f1 = 0.5144).

# Exercises



# Exercise 1

Consider a document collection consisting of 1M documents.

- What is the inverse document frequency of a term that occurs in every document in the collection?
- What is the inverse document frequency of a term that occurs in a single document in the collection?

(Compute logarithms with base 10.)

## Exercise 2

Consider two document collections  $A$  and  $B$  where  $B$  is obtained from  $A$  through lemmatisation. Suppose that the lexeme `RUN` occurs in  $A$  in the two inflected forms *runs* and *ran* (and no other forms).

Assume that we want to compute the tf-idf value of the word *run* in  $B$  based on quantities in  $A$ . What quantities exactly would we need to know for this?

## Exercise 3

Have a look at the documentation of scikit-learn's [TfidfVectorizer](#).

- What parameter is relevant if you want to use your own tokenizer, e.g. the tokenizer provided by spaCy?
- Explain how the parameters `stop_words` and `max_df` offer to different ways to filter out stop words. How do they differ?

## Exercise 4

Explain why information retrieval systems are typically not evaluated in terms of recall.

# Introduction to the lab

# Description of lab 1

- Your task in this lab is to implement the core of a minimalistic search engine for apps from the Google Play Store.
- More specifically, you will implement ranked search over a collection of app descriptions scraped from the Store.  
*tf-idf vectorization, cosine similarity*
- You are allowed to use a full set of Python libraries, including pandas, spaCy, and scikit-learn.

play.google.com

Private Research Teaching LiU

Google Play

match candies

Apps

Search Android apps All prices

My apps

Shop

Games

Family

Editors' Choice

Account

Payment methods

My subscriptions









Redeem

My wishlist

My Play activity

Parent Guide

### Apps

 <p>Candy Match Gökhan OKUMUŞ ★★★★★</p>	 <p>Sweet Candies 2 - Candy Match SmileyGamer Match 3 Games ★★★★★</p>	 <p>Candy Match Sugar PokutWold ★★★★★</p>	 <p>Sugar Candy Match Glue Games ★★★★★</p>
 <p>Candy Match - Free iJoyGame ★★★★★</p>	 <p>Neon Candy Match DeliciousGames ★★★★★</p>	 <p>Candy - Match Three match games blast ★★★★★</p>	 <p>Match 3 Candy 3583 Bytes ★★★★★</p>