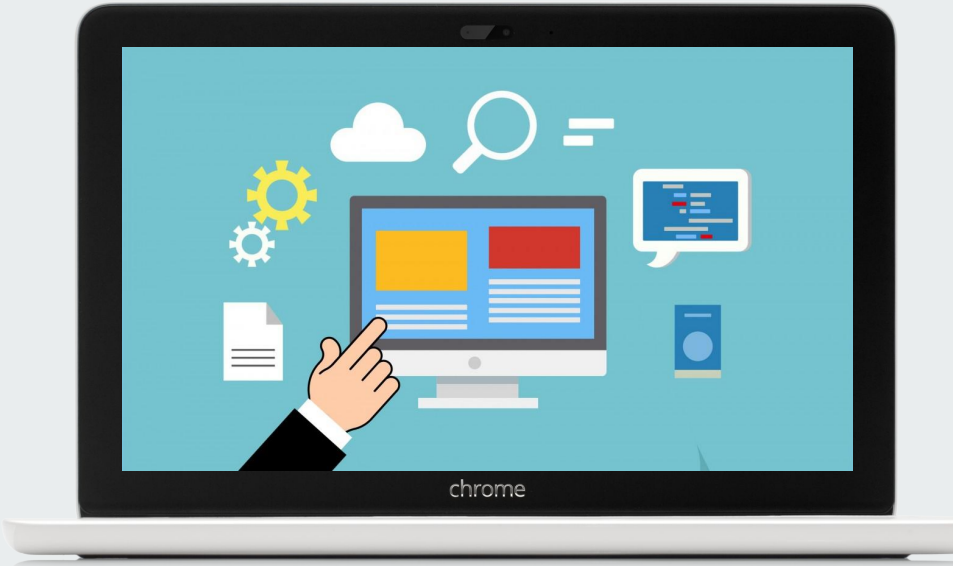
NLP for Software Engineering

Winter Term 20/21
Kickoff, 2020/10/27



http://wwwbroy.in.tum.de/lehre/seminare/WS2021/NLP/

Agenda

Organizational Things

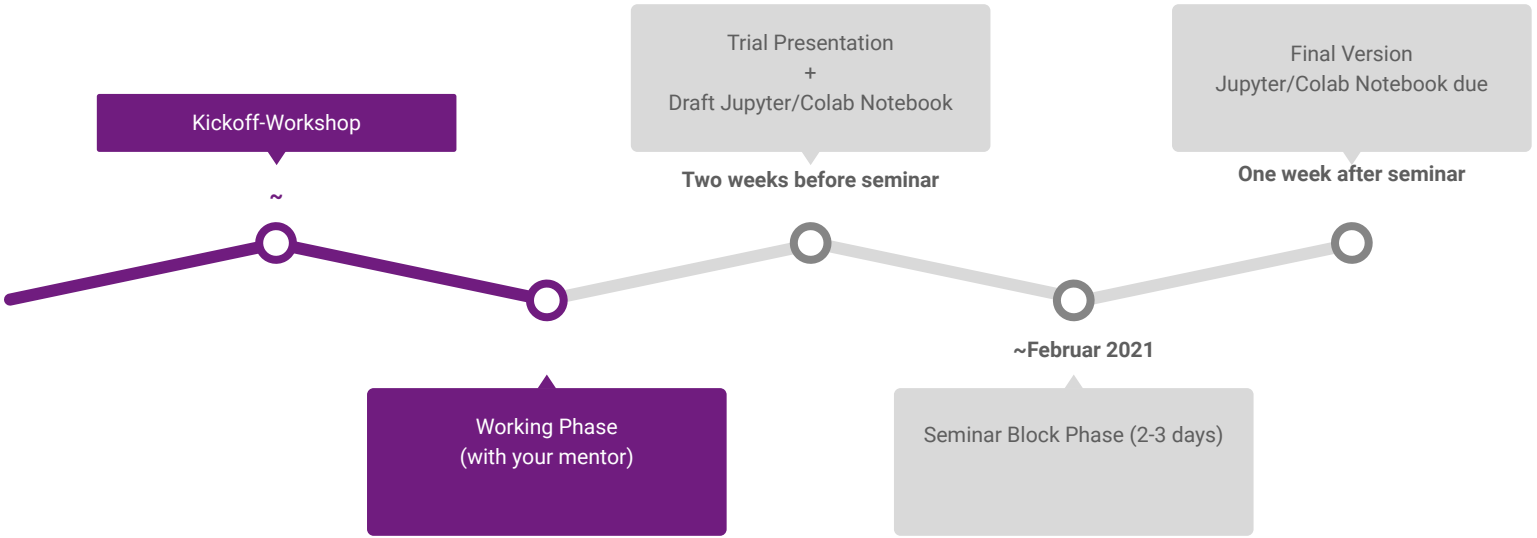
Presentation of Topics

How to write a thesis (in the NLP seminar)

How to do a literature review

Effective presentations

Reminder: Way through the seminar



Block Phase - Dates

Please fill out the Doodle

https://doodle.com/poll/q34gtebmkt7w6tdu

Collection of Results

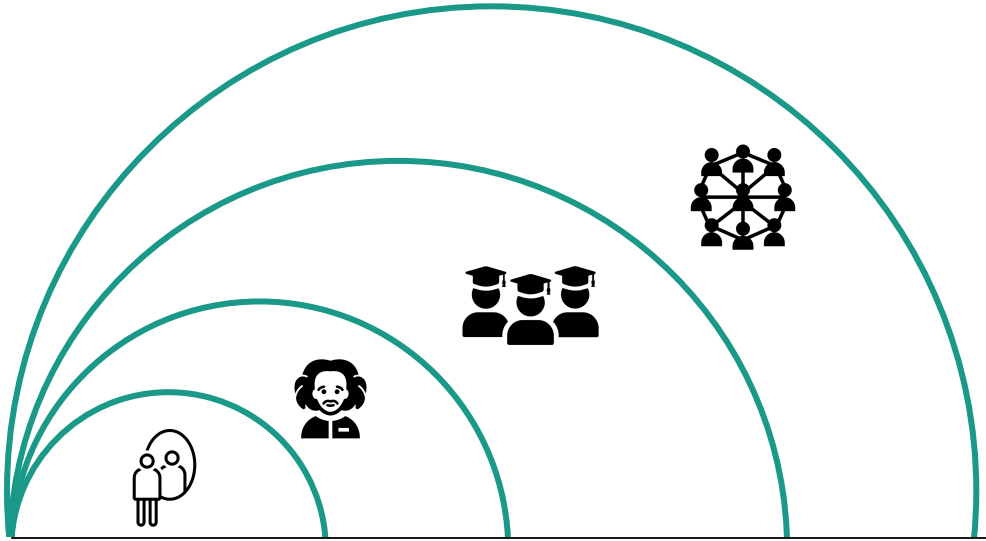
- Google Site to collect links to all results (Colab, Github, …)
- Add an image, a description and a link when your thesis is handed in
- Non-Mandatory
- Login to Site-Builder will be provided

Questions? Problems?

Presentation of Topics

How to write a thesis (In the NLP Seminar)

Who are you writing for?



- Yourself
- Your supervisor
- Your colleagues
- The scientific/technological community

Structure of a Thesis

1. **Introduction**
   a. Context
   b. Problem
   c. Contribution
   d. Outline
2. **Related Work**
   (Differences to existing work)
3. **Fundamentals**
   (Need-to-knows for understanding your text)
4. **Solution Approach**
5. **Evaluation**
6. **Summary**
7. **Outlook**

Some Thesis Writing Tips

- Start early with outline and introduction
- Clarify the high-level story
- Write fundamentals when everything else is finished
- “Shitty first draft” → Don’t aim for perfect formulation immediately
  (Start with just bullet points)
- Work with examples and illustrations
- Write in the “we” form
- In general: avoid passive voice
- Remember: You are not writing a novel
  - Use technical terms (even if that means repetition)
  - No need to aim for a surprising end
  - Write in present tense
- Favor simple language over complex formulations

Inhalt

Labeling causality in software requirements using a pre-trained language model

Seminar Natural Language Processing for Software Engineering, summer-term 2020

[redacted]16.08.2020

1. Introduction

Software-requirements can often be formulated in a form similar to "If A happens, B should happen". This concept, where the occurence of one event (cause) triggers another event (effect) is known as causality. According to Fischbach et al., 2020 [1], causal knowledge within software-requirements can be used to derive test-cases that control the correct implementation of functionality, for example "Check whether window Y is opened if the user presses button X", derived from the requirement "If user presses button X, window Y should be opened".

To automatically create test-cases from software-requirements, one needs to find a way to extract causality from natural language. The problem with causality extraction from natural language is that causality is embedded in very divers and variable language patterns and an algorithm would need a deeper understanding of language in general in order to recognize causality with certain accuracy.

In this notebook, I demonstrate how a pre-trained language model can be used to label causality in sentences. With my notebook, I want to show how accurate pre-trained language models can fulfill this task while also stating the weaknesses and problems of this approach.

2. Related Work

The problem of identifying cause-effect-relations in text has been subject to a few scientific papers. These papers can be splitted into two groups; rule-based approaches and machine learning approaches. Rule-based approaches use pre-defined language-patterns to extract causality and are very work-intense since all kind of causal patterns need to be defined manually in order to achive reasonable performance. [1], [2]

Other papers propose the usage of Machine Learning.
Rink et al., 2010 [3] train a Support Vector Machine classifier using contextual features in order to classify semantic relations which also includes causality. However, this approach only labels single words as cause and effect resulting in a loss of information that would be needed