729G17/729G86/TDP030 Language Technology (2023)

Project kick-off

Marco Kuhlmann
Department of Computer and Information Science

LINKÖPING
UNIVERSITY

Example projects from previous years

Detecting hate speech in tweets

- Build a classifier that can detect whether a tweet contains hateful speech or is offensive in some other way.

  datasets available on e.g. Kaggle

- Use several different document representations, including tf–idf (with several n-gram sizes) and word2vec.

- Explore a wide range of classification techniques, including support vector machines and neural networks.

  implemented in scikit-learn

Sentiment analysis of Twitter data

- Can we use text classification to predict the sentiment of a tweet in relation to a given topic?
- Build a 'silver standard' based on the hypothesis that :) indicates a positive tweet while :(indicates a negative tweet.

  innovative: noisy labels
- Collect data using the Twitter API, preprocess the data, train different text classifiers, identify most informative features.

Quantifying text emotiveness

- The notion of emotiveness refers to how emotionally engaged a writer or speaker was while producing a text.
- There are psycholinguistic theories about how emotiveness can be measured in text.

  Trager coefficient, aggressiveness coefficient, readiness to action
- Part-of-speech tag the inaugural speech corpus, analyse the emotiveness of the speeches over time, explain the results.

Implementation and analysis of a voice assistant

- Implement a voice assistant with Google's Dialogflow framework that can answer questions about student life at LiU.

  When is my next lecture? What computers are available?

- Evaluate via a user study to analyse what worked well and what type of issues occurred in conversations with the bot.

Change of language in Reddit over time

- Study changes in language on specified subreddits by visualising posts over a certain time span.
- Features used: sentiment analysis, average post and sentence length, verb/noun ratio.
- Relate the findings to events happening in the real world, including the days around Avicii's death.

Tips and tricks

Path 1: Start with the application

- One way to start the project is to pick an application that you find interesting and want to know more about.

  text classification, natural language generation, question answering

- Spend some time to find out what data sets and what software is available, and how systems are typically evaluated.

Path 2: Start with the data

- Another way to start the project is to pick a data set that you find interesting and want to know more about.
- Spend some time to actually look at the data. What have others done with it? What could you do with it?
- Be incremental. Collect ‘small’ results. Once you feel that you have enough, try to integrate them into a big picture.

How to get data?

- Ready-made datasets from shared tasks, data science competitions, public providers

  Kaggle, Riksdagens öppna data

- Data from companies made available via APIs

  Twitter, Musixmatch

- Scrape data using web scraping tools such as Scrapy

  may require preprocessing, manual annotation – licenses?

Implementation work

- Coding can be part of a project, but it is *not* the main focus.

  Remember the learning outcome!

- Do not start from scratch. Use existing software libraries.

  pandas, spaCy, NLTK, scikit-learn, Gensim

- Use whatever ecosystem you are most comfortable with.

  No requirement on the programming language.

How to validate?

- intrinsic evaluation using easy-to-calculate measures such as accuracy, precision, recall, perplexity, …
- extrinsic evaluation, for example by embedding the component into a larger system or doing a user study
- subjective evaluation of how easy it is to explain the results, how well the results fit the facts, how well they fit a theory

How to get help?

- Pitch your project idea to me!
- Project groups can book online meetings with me throughout the rest of the course.