



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Natural Language Processing and the Web Final Project

Dr. Seid Muhie Yimam and Saba Anwar

11/13 December 2024

This project is an opportunity for you to apply the theory and technologies you've learned so far to design and implement your own NLP system. Be creative!

Groups and grading

For this project you should work in groups of two to four. When you submit your project for grading, you must include a statement indicating which group members were responsible for which aspects of the project.

You need to pass the project to complete the practice class. Projects will be judged on several aspects:

- creativity
- methods used
- correctness of the code
- thoroughness and justification of your design decisions
- thoroughness of results analysis/testing
- quality of your project writeup and presentations

You will not be penalized if the results produced by your system aren't very good or very interesting, as long as your initial design decisions were sound, and as long as you made a reasonable attempt to analyze why your system failed and how it might be improved.

Scope and theme

You are free to make use of existing data, libraries, and source code for your project, provided that you document the source of all third-party resources and describe their benefit. Note that a project that does nothing more than run off-the-shelf components will not be enough to pass the practice class!

Your project *must* involve one of the NLP python frameworks we covered in class, such as NLTK, Spacy, TextBlob, Sklearn, Pytorch, LLMs, or any similar frameworks and it *must* be in the topic of NLP for the Web. The web data you work with can be in any language.

Timeline

11/13 December 2024 Project description distribution, ...

18/20 December 2024 Project proposal presentations.

8/10, 15/17, and 22/24 January 2025 In these practice classes we will make ourselves available for digital consultation and advice. Of course, on other days you are always free to contact us with questions by e-mail or on the discussion forum.

29/31 January 2025 Project report presentations. Deadline for submission of written project report.

Project pre-proposal

Please add the topic, small description, and group formation until **December 16, 2024** to the Moodle **"Final Project topic discussion"** forum **if you propose your OWN project idea**. This will not count towards your grade, but it is important to give us an idea of what you would like to work on so that we can provide help and feedback. For example:

- We will confirm whether what you propose is actually on-topic for this class. If it's not, we'll suggest how you might modify the project to make it more appropriate.
- We will tell you if the project you're proposing is too similar to what another group is doing. If it is too similar, we'll suggest how you might modify it.
- We will tell you if your project is too big or too small so that you can narrow or widen its scope.
- We will point you towards useful resources (data, software, documentation, previous research papers, *etc.*) that you can use.

The description should include the following:

- What research problem do you propose to investigate?
- What approach do you intend to take, including how you expect to evaluate your results.
- What resources you've found so far that you think will help. These can be things like software packages, data sources, and scientific or technical literature.
- What your work plan is, including who will be responsible for what and what sort of timeline you expect to complete various tasks.

You can also select the project topic from the suggestion **"Topic selection - first come first served - one member from team"**, which will be active on **December 16, 2024, at 10:00**.

Project proposal

You will present your proposal on 18/20 December 2024 during the practice class. This should take the form of a five-minute presentation with slides. The proposal should include the following:

- A clear statement of the research problem you will investigate.
- A clear statement of the approach you will be taking, including how you intend to evaluate the results, and your justification for choosing these methods.
- Which third-party resources you will be using and how you will be using them.
- Which resources you will be constructing yourself?
- Optionally, a brief report on any work you have already done.

Project report

Your project report is due on 29/31 January 2025. This report will be in two parts: a written component, which you will submit to us, and a presentation which you will give to the class.

Your written project report should be about five to ten pages long, and should generally include the following. (The exact structure and contents will vary from project to project; when in doubt please contact us for advice.)

- A clear statement of the research problem you investigated.
- A clear and complete explanation of the method you used to investigate the problem, including mention of third-party software you used.
- A high-level description of the software you implemented yourselves, including a class/component diagram.
- A discussion of the data sources you used and how you obtained and processed them.
- A discussion of any preliminary testing or training you did.
- A discussion of the things you tried to improve the system's performance, and how these worked out.
- Clear final results from running the system on real-world data. Don't just present raw data; use tables, graphs, diagrams, *etc.* as appropriate. Visualizations can be useful and interesting!
- A discussion and evaluation of the results. Were they the ones you expected? If not, why not?
- A discussion of future work which could be done.
- A statement indicating which group members did what.

You should also submit to us the full source code for your system, along with the raw output data (if it's not too large).

The presentation should be about ten minutes long and cover the same sort of information as the written report, though obviously not in as much detail. Try to highlight any particularly interesting results you may have achieved, using any appropriate visual aids.

Project ideas

Below is a list of project ideas you might consider. (Software, corpora, and other resources for them are given in the following section.) You are free to use or adapt any of them, or to come up with a completely different idea of your own. However, keep in mind that each group must work on projects which are at least slightly different. In case two groups submit the same project idea, whoever submitted theirs first will have priority.

Another good source of ideas is recent research papers in NLP. Many conference papers are available online in the ACL Anthology at <http://aclweb.org/anthology-new/>.

- A1 **LLM for Legal advisory**: Using LLM and retrieval augmented generation approach (as we cover in the last practice class using **LangChain**), build a legal advisory system, particularly filling a form. This might involve scrapping legal documents from the Web. **If you are interested in this project, contact us, we have a project from another student that you could extend it**
- A2 **ACL Anthology for NLP**: The ACL Anthology GtiHub repository contains: 1) Metadata for all papers, authors, and venues on the ACL Anthology website. 2) Code and instructions for generating the website. 3) A Python package for accessing the metadata, also available on PyPI. Using this repository, you can do some of the following:
- (a) Find a similar paper. This will help especially in researching similar topics. It can also be used to cite appropriate works for parts of the text you are writing (there is the Bibtex file associated with each paper). You can use RAG to search for similar papers, find relevant topics from the database, and cite works.

- (b) Designing a plagiarism detection system. This repository can help in detecting and finding relevant sections that are copied verbatim. You can incorporate scoring of the similarity, identify/highlight similar content, and so on. You can use the RAG system.
- A3 LLM for Tabular Data Analysis :** Follow [This tutorial](#) and build QA system over tabular data. You can choose one of the following:
- (a) Financial Document: For example, use this [MERCK Report](#) or more of the [MERCK reports from here](#) to ask and get answers from the tabular data. Design your system so that it can also help kind of questions you can ask over the datasets.
 - (b) Medical document: Use the [The Global Health Observatory - the world of health data](#) and allow asking any question regarding the dataset. Design your system so that it can help you understand about the dataset, kinds of questions to ask, the topics and so on.
- A4 Hallucination-detection :** From this list of [awesome-hallucination-detection](#) projects, try to implement one project related to hallucination detection and evaluation.
- A5 Academic Recommender:** It is very challenging to find the right person, for example for a collaborative research project, finding the right professor for supervision, or even, suggesting a common research topic for researchers with a similar research profile. Can you develop a system, that could also use LLM, which provides the right academic person group based on existing research publication repositories such as OpenAlex or Google Scholar?
- A6 Stackoverflow Question Answering:** Develop an NLP system for QA, using Stackoverflow (<https://stackoverflow.com/>) data archive, in its current state, it provides a list of questions most similar to user asked the question, with answers and other metadata (tags, votes, views, etc). Given a user question, your system should provide the answers ranked from most relevant to least, making the answers more recent would also be a nice feature.
- A7 Stance detection:** Develop an NLP system for stance detection. a) You can scrap data from <https://debates.org>. Given a question or statement, the website provides a list of user stances both in favor (YES) and against (NO) the statement. Optionally, your system should summarize the whole debate briefly. b) Or, use the dataset from here: <https://github.com/ZurichNLP/xstance/tree/master/data>
- A8 Explainable Opinion** Use existing datasets for sentiment analysis, opinion mining, and emotion analysis and explore how to explain the model output. For example, if the label of a tweet is negative, why is it so? You can start with keywords or lexicon. You can employ the attention network from BERT-like transformer models.
- A9 NLP for social good:** In this project, the main purpose is to use unstructured information available online, for example from <https://www.hamburg.de/>, and build more advanced NLP tool that could serve information retrieval needs. You might need to apply web scrapping or ask the Hamburg city if they can provide an archival.
- A10 NLP and Finance :** Build an application for financial sentiment. This is different from traditional sentiment analysis task as the main goal is to assess how the market react based on a given event. You can check FinBERT (<https://arxiv.org/abs/1908.10063>) and adapt it to live texts, for example texts collected from Twitter that target a given company.
- A11 Fake News and Disinformation:** The LT Group participated to the "Hackathon on COVID-19 Related Disinformation (Virtual Event)". The event was mainly targeting to build solutions in an effort to debunk fake news. The LT group has participated with the EUvsDisInfo dataset but we have access to all of the datasets provided at the Hackaton. The result of the Hackaton from the LT group can be found here¹. Either enhance one of the components in the pipeline developed by the LT group or propose a solution for the other datasets.
- A12 Amazon Customer Reviews:** This is a dataset about Amazon customers or products. This a recent and rich set of data source openly available for academic research. The dataset contains reviews from 1995 to 2015. One option is to predict stars (1–5) using review texts. Details about the dataset is available here <https://s3.amazonaws.com/amazon-reviews-pds/readme.html>

¹https://docs.google.com/presentation/d/1Rj5vQ8DV4ed6gxFENk-mnqWfVbYW2_cNBmWlq7LLjg/edit?usp=sharing

A13 **NLP in Social Science:** You can pick a topic related to social science such as deception detection (see different topics here <https://web.eecs.umich.edu/~mihalcea/downloads.html>) or fake news classification (see for a data here <https://github.com/JasonKessler/fakeout>).

A14 Develop a system which does **information extraction** from the text of online recipes. For instance, can your system identify all the ingredients and their amounts, what actions are taken on the ingredients (*e.g.*, mixing, boiling, washing), any new entities produced by these actions (*e.g.*, mixtures), the containers used to hold ingredients, and/or the utensils and tools used to process the ingredients? Alternatively (or additionally), can your system identify ingredients even when the directions use different words for them than the ones used in the ingredients list? (For example, the ingredients might list “almonds” and “walnuts” separately, but the directions might refer to them together as “the nuts”.)

For this project you will need access to an online recipe database. There are many websites you may be able to scrape. LT can provide access to chefkoch.de data but still you have to deal with html parsing and stripping yourself.

A15 Construct a multiple-choice question-answering system which could compete on the television show **Who Wants to be a Millionaire?**. You could use web data to solve the task (Wikipedia might be a good one) or make use of (for example) the Yahoo! Answers API. LT might provide access to English or German Wikipedia instances.

... and some more

B1 Conduct a study involving **sentiment analysis/opinion mining of social networking or microblogging posts**. For example, you could determine how to track a product or public figure’s popularity over time, and possibly also correlate this to real-world events reported in the news media or blogosphere. This project has great potential for nifty visualizations.

For this project you will need access to a corpus of social networking or microblogging posts. You could construct this yourself (using, for example, the Twitter API, or producing your own Facebook app) or you might be able to use one of the publicly available Twitter or social media corpora.

B2 Construct a system for **author classification of blogs**. For example, can you build a system which reliably determines the gender, age, political affiliation, or other biographical information of an author from his or her blog posts? For this task you will need a corpus of blog posts such as those in the ICWSM datasets.

B3 Write a **text summarization system for a particular domain of online reviews**. For example, review sites like Rotten Tomatoes and Metacritic present condensed summaries of critics’ movie reviews. How could you produce something similar by extracting the sentences which best capture the opinion of the reviewer? Or alternatively, could you build a system which looks at a number of reviews and then identifies and summarizes the most common praises and criticisms, without duplication? For this task you will need a corpus of (possibly marked-up) reviews, such as movie reviews or product reviews.

Projects from previous years

1. Sentiment analysis on Reddit comments: **Approach** choose one subject, crawl Reddit comments using an API, Use Twitter Sentiment Analysis dataset to train a model, User study to check the opinion of the result.
2. Sentiment analysis on Twitter: **Proposal:** Perform a sentiment analysis on Twitter Data regarding the topic Donald Trump **Aim:** See how accurate the Stanford Sentiment Analyzer is on Twitter Data and whether it’s possible to create a mood picture about Donald Trump by the collected Data.
3. Automatic Identification of Cause-Effect Relations with UIMA: **Goals:** Identifying cause-effect relations with UIMA, Problem is based on the Milton Model by Milton Erickson. **Approaches:** Keywords that indicate cause effects: because of, thus, unless, since, therefore, in order to and so on. . . – Part of speech types: NP, NN, ADJP and ADJ – **Underlying patterns:** cause - keyword - effect, effect - keyword – cause , (keyword - cause – effect).

4. Generate Graphs from Online Recipes: **Motivation:** Online recipes are often big blocks of text, Cooking multiple recipes in parallel, Using Flowcharts for a clearer and faster view
5. Predict Star Ratings for restaurant reviews: predict the numerical rating, uses ML techniques to predict star ratings, Evaluate the programs with reviews from different domains
6. Tweet-based political sentiment analysis: identification of German politicians in tweets, automatic sentiment analysis with ML, evaluate with official sentiment surveys and visualization.
7. "Who wants to be a Millionaire" questions using Wikipedia, Google, and Bing: extract relevant words from the question, use the web as corpus, output is an answer and a confidence score
8. Tag-Prediction in stackoverflow: collect questions with tags from stackoverflow, apply ML, used Named entity, Token and N-Gram as features
9. Use network of the day (NoD) to visualize entity networks for other languages - Access NoD from here <http://www.tagesnetzwerk.de/> . The tool main display entities (Person and Organization) linked each other when they have relationships. The group last year extend NoD to English Language. They Download news feed for English using RSS feed aggregation, apply machine learning to extract entities and use a simple heuristics to extract relationships (two entities occurred in the same sentence holds relationship). To read RSS feeds, they use an existing parser such as horross (<https://code.google.com/p/horross/>).
10. Step detection in recipes: The main goal of the project is the identification and extraction of steps from a recipe. The scope is limited to the culinary sphere so the set of words of interests is also limited.
11. Sentiment Analysis of TripAdvisor Hotel Reviews: Given a set of labelled hotel reviews from TripAdvisor build a classifier, that would predict the rating given to this hotel based on the textual part of the review and the meta-data.

Useful resources

Here is a list of resources that might be useful for getting ideas for a project, or for your project itself. Links to additional resources may appear on the course's Moodle page; you can also find more yourself with some web searches.

Corpora, lexical resources, and other data sets

10% of Stack Overflow Q&A

<https://www.kaggle.com/datasets/stackoverflow/stacksample/data>

Stackoverflow data links

<https://www.kaggle.com/datasets/stackoverflow/stackoverflow>

Twitter small corpus

<http://infolab.tamu.edu/resources/>

1000 randomly selected twitter users and their tweets.

Tweets2011 - 2012

<http://trec.nist.gov/data/tweets/>

16 million tweets from 2011. Note that it can take several days to obtain this corpus, as you must first submit a data usage agreement. A similar corpus in JSON format is available on request from LT lab.

WordNet

<http://wordnet.princeton.edu/>

A lexical database of English which records semantic relationships between words.

Squirrel's RecipeML Archive

<http://dsquirrel.tripod.com/recipe/ml/indexrecipes2.html>

10 000 recipes partially marked up in XML.

ICWSM data sets

<http://icwsm.org/data/index.php>

Various annotated and unannotated corpora covering blog posts, social media, forum posts, classified ads, reviews, Wikipedia contributions, personal stories, *etc.* Note that some of the corpora require you to submit a data usage agreement which can take several days to process.

Movie Review Data

<http://www.cs.cornell.edu/people/pabo/movie-review-data/>

Tens of thousands of online movie reviews, thousands of which have been annotated for sentiment and subjectivity.

Multi-domain Sentiment Dataset

<http://www.cs.jhu.edu/~mdredze/datasets/sentiment/>

Hundreds of thousands of product reviews from Amazon.com, annotated by product type and star rating.

The 4 Universities Data Set

<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/>

8282 university web pages manually classified into categories.

WaCKy

<http://wacky.sslmit.unibo.it/doku.php?id=corpora>

A number of annotated and unannotated multibillion-word web corpora.

Leipzig Corpora Collection

<http://corpora.uni-leipzig.de/download.html>

Millions of randomly selected sentences from the web.

Wikimedia dumps

<http://dumps.wikimedia.org/>

Downloadable versions of Wikipedia and other Wikimedia projects, in HTML, XML, or raw SQL tables.

Statistical natural language processing and corpus-based computational linguistics: An annotated list of

<http://nlp.stanford.edu/links/statnlp.html#Corpora>

Links to various other corpora.

PPDB: The Paraphrase Database

<http://www.cis.upenn.edu/~ccb/ppdb/>

Paraphrases developed from parallel corpora. PPDB for English database access can be available at LT lab.

RSS feeds

BBC: <http://feeds.bbc.co.uk/news/rss.xml?edition=uk>, <http://feeds.bbc.co.uk/news/rss.xml?edition=us>, <http://feeds.bbc.co.uk/news/rss.xml?edition=int>, Reuters News RSS Feeds lists - scrap from here: <http://www.reuters.com/tools/rss> Fox News, scrap from here: <http://www.foxnews.com/story/2007/11/09/foxnewscom-rss-feeds/>

chefkoch.de data

Dataset downloaded from chefkoch.de and available with metadata in RDF quads

TripAdvisor Data Set

<http://times.cs.uiuc.edu/~wang296/Data/>

Reviews crawled from TripAdvisor. Meta data includes: Author, Content, Date, Number of Reader, Number of Helpful Judgment, Overall rating, Value aspect rating, Rooms aspect rating, Location aspect rating, Cleanliness aspect rating, Check in/front desk aspect rating, Service aspect rating and Business Service aspect rating. Ratings ranges from 0 to 5 stars, and -1 indicates this aspect rating is missing in the original html file.

Software and APIs

Twitter API

<https://dev.twitter.com/>

Facebook API

<https://developers.facebook.com/>

Yahoo! Answers API

<http://developer.yahoo.com/answers/>

TextRunner

<http://www.cs.washington.edu/research/textrunner/>

JWPL (Java Wikipedia Library)

<http://www.ukp.tu-darmstadt.de/software/jwpl>

An API for Wikipedia which can operate on the above-noted Wikimedia dumps.