PM: Mining Opinions and Arguments
University of Potsdam, Winter 2021/22

Term Project Guidelines

Original version: Tatjana Scheffler (2018)
Adapted by: Manfred Stede

1 Overview

Task. The grade of the course is determined based on a **term project**, which should implement a system related to sentiment analysis and/or argument mining. Usually, the task is based on some research paper/s and is concerned with reproduction of earlier results and extending the approach with own ideas. Some variants of this general theme are possible; the specific tasks are negotiated in weeks 4/5 of the seminar. Any project must have an *evaluation* component that is suitable for the type of task. The project is to be documented in an ACL-style paper that adheres to the standards of practice (form, content) in computational linguistics.

Groups. The term project can be carried out alone or in groups. If it is a group project, all members are assumed to be involved in all parts of the project and will receive the same grade. Should you run into problems with this default rule, i.e., if you think different grades should be assigned to different subprojects (= students), contact the instructor asap.

Content and Scope. At 12 LP the class should take up almost half of your workload this entire semester (considering full time studies). Following the overview lectures of sentiment analysis and argument mining, the instructor will suggest a list of candidate topics for the projects. Proposing your own topic is possible, but relatedness to those on the candidate list is desirable. Once the topic is assigned and specified to sufficient detail, do the required reading and begin carrying out experiments. This could be your own implementation, but could also be adapting or evaluating existing implementations. It is important that you delve deep into at least one aspect: That is, don't only try one thing and declare it done, but also try to *understand* why it worked/didn't work, and try some things to improve it.

The final **goal** of your project depends to a good extent on the type of task (and will be negotiated). Generally, it is not expected to produce new international state-of-the-art results on your task (but if things work out this way, don't feel bad). The point of the project is to really know your way around the task and the data, to get your hands dirty with the necessary implementation

work, and achieve satisfactory quantitative results, which in turn presupposes carrying out a reasonable evaluation of your work. And, of course, all these things rest upon organizing a fair division of labour in the group.

Grading. Once again, to some extent the grading policy depends on the type of your task that we negotiated. For example, when the bulk of the work is a difficult reproduction of a published approach, the first criterion below is of small significance. Anyway, generally speaking, the grade takes the following criteria into account:

creativity How interesting and well-motivated is your topic/research question? Is it standard or did you come up with it and defined your own task cleverly?

effort How much own work went into your project? Was there a lot of manual/preparatory work, e.g. to create a dataset or annotate, or to get some code running, or a lot to implement? Did you only try the simplest things or the ones that come to mind first, or did you try hard to improve some aspect?

understanding/correctness Does the approach you chose make sense for the task? Did it work, or, if it turned out to be a blind alley, were there, anyway, good reasons to try it? (Negative results are not by definition bad!) Are there any problems in conceptualizing the problem, and mistakes in your understanding of basic issues? Is anything you wrote in your report blatantly wrong? Are the algorithms/methods described correctly?

argumentation/writing/structure Is there a coherent thread throughout the paper from the motivation through approach to results? Can the reader immediately follow what you did and why? Is the writing clear and concise?

formal Is the paper proofread? Do you adhere to academic standards by citing relevant papers and comparing to them? Is the formatting consistent and does it make the paper easy to read? Do you include tables and graphs and refer to them in the text?

graphs/code Do graphs and tables support your arguments / display your results? Can your code be run? Is your code documented and well-structured, so it can be understood and run/adapted by someone who has read your paper (or you yourself in one year)?

quality of semi-final presentation in February (see below)

These criteria are not necessarily weighted equally, but they will all influence your final grade.

Milestones and Requirements.

Oct/Nov Introductory lectures. Assignment of topics; specifics negotiated individually with the groups.

December Groups present brief project plan (and possibly first steps) to the class.

February Groups present semi-final results to the class.

March 31st Final submission deadline for code and report.

April Optional: final project presentations (not graded)

Between December and February, there are individual consultations of teams with the instructor. We usually use the standard seminar time slot; dates to be negotiated.

2 Project presentations

Plan presentation. In December, each group briefly presents their project idea and approach. This will also let you know what the others are doing and enable you to give each other feedback. Each presentation should be no longer than 10-15 minutes, and you may designate one presenter or split it up among you. You provide a clear motivation and problem statement, a description of your approach, and your plan on how to evaluate your project. It is typically necessary to refer to previous work as well, in order to provide context, background, and ideas for what works and doesn't work. Depending on how things are going, the presentation can include some preliminary results of yours (mabe a baseline).

While the others are presenting, think of some questions or comments to contribute! This plan presentation is not graded.

Semi-final presentation. Groups present their topic and research question (as a brief reminder for the audience), motivation, methods and approach, as well as the results achieved so far. This presentation takes roundabout 15-20 minutes, plus there will be time for questions and discussion. A presentation can conclude with an outline of the final steps to be taken, and with points you see as fruitful to be discussed with the audience. You can also include a demo if one is available already (and if a demo is indeed illustrative for your type of project).

This presentation will be part of your grade (coherence and completeness of content, clarity of slides and oral presentation).

Optional final presentation. We try to find a time slot in late April where groups informally present their final results, and swap thoughts on how things went.

The point of this presentation is to update everybody about the progress made and lessons learnt in the final weeks. It is not graded.

3 Project implementation

The final project should be handed in as documented, running Python code. For larger projects, a download link (e.g. to Github, my id is `mstede`) suffices. Please also provide instructions on running the code (including a requirements.txt) and

thus verifying the results. This documentation, though it can be brief, is an important criterion for your submission.

4 Project report

The project report is in the style of an academic paper that documents your research questions, data sets (if any), methods and results. Please use a current ACL/COLING/EMNLP style file for Word or LaTeX, and write your document so that it "looks and feels" like a standard ACL paper. We are not terriby strict about length requirements, but aim for a customary 8-page paper (excluding references) in the ACL style. Appendices can be used for longer examples, extra tables and figures, and the like, in case you have more material that should be presented.

Importantly, the project report should stand on its own, i.e., it should not assume that its readers are already familiar with your project, and so it definitely needs a good "introduction" section. And, as it should look like an ACL paper, also a short abstract.

Academic Writing. The papers you submit are part of a specific genre, academic writing. Student papers are special since they serve the purpose of clearly documenting what you did (as in scientific papers), but also to demonstrate what you know/have learned. Please remember that your instructors can only grade you based on what's written in your papers – so try to make it easy on us to understand what you know and what you have achieved. Initially, it can be hard to judge which things are "obvious" and shouldn't be explained and which things need repeating (even though we both know them). It usually helps to picture an intelligent peer who *hasn't taken this class yet* as your target audience. I.e., you can assume most basic knowledge about the field (of the kind that undergraduates possess) as given, but should explain or at least mention anything that is beyond that level, though briefly.

In terms of style, academic writing is very results-oriented, focusing on outcomes and argumentation much more than on the process. The most common mistake by students with little university-level writing experience is to faithfully narrate what they did (and how they changed their approach in the course of the project) in largely temporal order. Instead, try to focus on methods and results – it matters much less how you got there, especially when you don't have much space. I recommend stating your main result early on (in the introduction).

All non-obvious statements need evidence or even proof, either by argumentation (why did you conclude this?) or by citation. Make sure to cite appropriate literature. It is clear that you won't have time for a thorough literature review, but in many cases there are straightforward previous approaches, and if you rely on them, you must cite them correctly. The same applies to code packages and data sets that you use in your project.

Use LaTeX if you can. I recommend Overleaf (`https://www.overleaf.com/`), a web-based LaTeX editor, no installation required.