

GRASP HOMETASK

You find yourself in the following situation.

Back in 2015, you published results to compete at [SemEval-2015 Task 1: Paraphrase and Semantic Similarity in Twitter \(PIT\)](#). Your model did especially well at the Semantic Similarity part of the challenge.

Because of this, the social media company in question approached you this week. They want you to design and build a system which moderates their user generated content relative to whatever their current code of compliance is. The code of compliance explicitly bans certain language (discriminatory, abusive, etc.) and is constantly changing over time. The company must be able to justify to a regulator why they made each moderation decision, whether correct or incorrect. The regulator dislikes false positives and false negatives equally. The company want your system to moderate all their traffic, and this traffic is expected to grow over time.

You are surprisingly excited by this and will do everything it takes to get the job!

You are told by someone who knows the company well, that doing the following exercises would convince them to hire you:

- Exercise 1: You make a new proof-of-concept model, which competes at SemEval-2015 Task 1 and provide an output of your performance in the same format (see `scripts/pit2015_eval_single.py`)
 - You document your thought process for each major design decision.
 - You assume that the work will be reviewed by a technical product owner.
 - Your model design and results should demonstrate that you are up to date with relevant technological advances since 2015, allowing you to pull something good together in a short period of time.
- Exercise 2: Detail how you would approach the company’s problem under 2 scenarios. You want to show the company that you can achieve a lot with a small amount of resources, but also want to propose what could be achieved with more resources.
 - Scenarios:
 - * (A) Yourself, 2 days, part-time, scarce resources (you have your laptop, the internet, this repo)
 - * (B) The resources you would need to do this really well (say what you need).
 - Guidance:
 - * Showcase how you can utilise your model from part 1, or a derivative thereof.
 - * Explain any additional work or resources (data, hardware, people etc.) you need
 - * You can use your resources for anything legal.

- Exercise 3: Implement your proposal for Exercise 2A

Task

Complete as many exercises as possible, in ascending order.

Submission

- Create a **private** github repository
- Push all this code there via git
- Commit and push all your changes to the repo via git, including:
 - a SUBMISSION file (e.g. txt, md, link to google doc) with overview of your work, what progress you made in the time frame, what the next steps would be, any relevant reference documents/links etc.
 - Any code
 - Any trained model (or link to)
 - Script to generate model predictions on test data, in /scripts
 - Output of model predictions on test data, in /systemoutputs
 - Txt of evaluation statistics on your data, per scripts/pit2015_eval_single.py, and your interpretation of these statistics.
- Share the repository with the emails of your contacts at Grasp.
- Setup a call with Grasp to discuss your work.

Considerations

We will evaluate your submission on both a qualitative and quantitative basis.

* Qualitative: interpretability of work, reasoning for decisions and roadmap etc, practicality for a startup. * Quantitative: the performance of your model relative to Table 3 of [SemEval-2015 Task 1](#).

Misc

- We are most interested in understanding your thought process throughout the project. Please annotate your work with notes or diagrams to illustrate what you're doing.
- We are agnostic as to whether you adapt off-the-shelf models or build from scratch, but we want to know why you take the decisions you take.
- This repository contains all the original files from the [SemEval official codebase](#), adapted by us from Python 2.x to 3.x for you.
- If you want to run 'scripts/baseline_logisticregression.py' (unessential to this home task) you will need to install [megam](#) on your machine for use with [nlTK](#). If you are on OSX, this is quite fiddly. However, should you be so inclined, you will probably want to:
 - install ocaml with brew
 - symlink ocaml/libbigarray.a to ocaml/bigarray.a

- Consider the discussion in [this thread](#). Again, we do not advise running it, it is unessential to the home task as the outputs are already provided.
- Any questions, please email.