**Chatbot assignment**

Deadline: March 22nd (~~Monday~~)

## *Goal:*

The goal of this project is to write a working chatbot by applying some of the techniques that we have seen (or will see) thoughout the course. This means that your chatbot should not only look for predefined keywords and use them to query a database but should rather implement, at least, one more "intelligent" feature (such as synonym detection using Word2Vec, text generation, sentiment analysis, formal-informal speech detection, etc.). You might well not even set up a database.

Your chatbot should be able to:

- Do some basic chit-chat: greetings, introduction (of abilities) and goodbye.
- Answer at least two types of domain specific questions. For example, if you chose to focus on movies, your bot might be able to give a quote from a given actor (or movie), tell you when an actor was born, or list a director's films.
- Do something else than just querying a database. Use any NLP technique, such as Word2Vec, to try and extend your functionality.

I'd like the whole thing to be turned in as a GitHub repo, with a complete README.md describing what you've done and giving an example dialogue demoing it.

The use of any external API is allowed. For example, you could use an API to retrieve information about the train services (https://www.ns.nl/reisinformatie/ns-api) or the weather (https://openweathermap.org/api).

## *Proposed work plan:*

For week 1, I recommend starting by getting a Telegram bot set up, playing with the basic Markov Chain text generator (markov_norder.py), deciding on your basic domain, coming up with a list of possible query types, and look for a dataset.

For week 2, you should be implementing at least some basic queries (i.e. mostly substitution-based), and starting to integrate smarter functions (e.g., similarity-based substitution for recognition or answering).

In week 3, you should finish your remaining functionality and try to improve usability if possible: have some people use it and see what modifications are needed to make it more

natural.

## *Resources*

**Chatbot ideas and information**

http://www.nltk.org/api/nltk.chat.html pattern matching bots in the NLTK api

http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/ basic concepts: retrieval vs. generation

http://www.wildml.com/2016/07/deep-learning-for-chatbots-2-retrieval-based-model-tensorflow/ LSTM-based retrieval system

**Corpora**

http://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html
https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment/
https://machinelearningmastery.com/datasets-natural-language-processing/
Any of the datasets included in Blackboard.
The presidential speeches included here.

**Scrapping resources**

Need some more data? Web scraping resources1:
https://first-web-scraper.readthedocs.io/en/latest/#act-3-web-scraping
https://doc.scrapy.org/en/latest/intro/tutorial.html

There are now actually good scraping tutorials, libraries (rvest), and text manipulation libraries (tidytext) for R, too:
https://blog.rstudio.org/2014/11/24/rvest-easy-web-scraping-with-r/