Suggested project topics

If none of the topics below seem tempting, you are very welcome to suggest a project topic of your own by e-mailing Johan.

Implement a word predictor

Most modern mobile phones have some kind of word prediction software. As you are typing a word in a text message or an e-mail, the system displays a list of the most probable completions of the word. The list of suggestions is updated for each keystroke the user makes. If one of the suggestions indeed is the word you intended to type, you can type the word just by clicking on it, thereby saving many keystrokes. A good word predictor would use word n-gram probabilies to suggest the most probable completion of the typed letters, and/or learn new words that the user is typing, and/or suggest corrections in case the user seems to have misspelt. Exactly how your system should work is up to you.

Text classification

Solve some text classication problem using scikit-learn, Weka, or some other publically available machine learning library). The problem could be author identification, genre identification, spam identification, sentiment analysis or similar. Collect a suitable corpus of data, and experiment with different methods, parameters, and ways to prepare the input. Evaluate your results.

Political language

In this variant of the text classication problem, we want to see whether we can predict the political colour of, say, the speaker of a parliament debate, or the author of a newspaper editorial or op-ed. Sprakbanken has plenty of material.(**https://spraakbanken.gu.se/swe/resurser/corpus (https://spraakbanken.gu.se/swe/resurser/corpus)**).

Twitter language

Do something interesting with tweets, e.g. try to predict the hashtags present in the tweet, or analyse the sentiment in the tweet.

Number classification

Write a program that classifies numbers in running text. Possible classes are age, distance, date, time, quantity, money, reference (e.g. "chapter 3"), etc. An interesting task is to compare a machine-learning approach with a simpler approach based on regular expressions.

Punctuation predictor

Remove all punctuation (!?,.) from a text, and lowercase all letters. Now write a program that puts the punctuation marks back again as well as possible.

Gender identification

Are men mentioned in the news more often than women? Write a program that counts the number of mentions of men and women, respectively, in an article. The program should have some strategy of dealing with names, and expressions like "the president" and "secretary Pompeo". This task can be made more difficult by considering texts in another language (like French), where pronouns don't necessarily refer to people, or by letting the program collect texts from the web.

Improved direct translation

Write a program that translates word-by-word from one language to another (e.g. by using **http://bab.la/ (http://bab.la/)**). Then improve your translation using n-gram probabilities for the target language.

Extract character descriptions from the full text of a book

Find out relationships between adjectives and characters, and between first and last names, between titles and first and last names, etc. Try to generate short summaries (using generation templates) of every character and compare the generated descriptions to human-written character summaries. English books in the public domain to use for this (e.g. Shakespeare, Jane Austen, the Bible) can be found at **Project Gutenberg (https://www.gutenberg.org/)** .

Harry Potter generation

Generate Harry Potter-like text by training a text generator on Harry Potter books (or the books of some other prolific author). Use Recurrent Neural Networks or Hidden Markov Models. See if a neutral judge can tell real Harry Potter sentences/paragraphs from fake ones.

Eurovision lyrics generation

Create a model that generates lyrics in the style of Eurovision songs for a specific country. Here is some data: **http://4lyrics.eu/eurovision/** **(http://4lyrics.eu/eurovision/)** (Obviously some other music genre would do just as well). How would you evaluate the results?

Folk Tune Title Generator

The application at **https://folkrnn.org/** **(https://folkrnn.org/)** is a music generation system trained on 23,000+ folk tunes from Ireland and the UK. An important thing this system does not generate is a title. The objective of this project is to create a system that will produce a title for a given piece of music. For instance, here's a tune from the training data:
M: 4/4
K: Amin
|: eA (3AAA c2(c2|c)Acd e2dc|eA (3AAA ABcA|BGdG eGef|
eA(3AAA c2(c2|c)Acd e2dB|GdBd e2dB|e2dB eA (3AAA:|
|: eaag a2ga|a2ga c'bag|a2ga eage|dged Bde2|
eaag a2ga|a2ga c'bac'|ba^gb a^ge^f|^gedB e2ed:|
The real title that goes with this tune is, "Granny Hold The Candle While I Shave The Chicken's Lip".