Exam Specifications

For students of B.Eng.602 (formerly B.EP.301) only, the course is assessed by way of a written exam taking place in our regular classroom on Tuesday 14 February 2023 from 10:00 to (a little after) 11:00am.

Remember that you will have to sign up separately for this exam and the lecture exam, which will also be administered separately.

The seminar exam will **not** ask you to do any actual programming. Instead, expect a mix of "how-would-you"-type questions and theory-based questions (define, explain). There will be one question asking you to describe premodern NLP work you would undertake if you had the chance, so do some research on existing projects and corpora and prepare a concise project proposal ahead of time. More generally, come prepared to discuss topics like:

- The concepts and vocabulary of NLP Python programming: lists, dictionaries, libraries/modules, functions, variables, imports, for-loops, indices, tokens, types, corpus readers. *(Low priority)*
- Lexical diversity, TF-IDF, Zipf's Law, cosine similarity, word embeddings, latent semantic analysis, sentiment analysis: what are these concepts and methods, how do they work, what are their strengths and limitations?
- Tokenization, normalization, stemming, lemmatization: what are these processes, what are some of the most general approaches to them, what makes them hard?
- Word sense disambiguation: how would you describe this challenge, and what can the available approaches contribute to its solution?
- What tasks are best tackled using machine learning, and what tasks are better served with a rule-based approach? How may both approaches be combined in e.g. lemmatization?
- Topics in NLP: which are the typical tasks that are tackled in practical NLP, what can they be used for, which ones are easy, which are hard, and why?
- What are some of the differences between NLP tasks for different modern and premodern languages in view of the natures of the languages, the tools available, and the goals of the work?
- Where do you see room for potential in the computer-based analysis of premodern language corpora? What approach would you apply to what data set if you were an NLP researcher?

How to Prepare

- **Class notes**: Class discussion is a central guide to what to expect on the exam. If it's not picked up in class, it's considerably less likely to make it onto the exam; if it's brought up time and again, that increases the odds it'll make it onto the exam.
- **Syllabus study questions**: Any of the study questions posed in the syllabus may appear on the exam, either verbatim or adapted.
- **Slides**: Some of our slides (but not all) will be highly relevant. Look them over, pay particular attention to the ones that shed light on concepts you encountered in your readings.
- **Textbooks/readings**:
    1. **Bird et al.** has been our foundational textbook and our easiest text, so it should also play a key role in exam preparation. Since there will not be a programming task on the exam, ensure you've understood the concepts, logic, and principles introduced in the book.
    2. **Lane et al.** represents some of the conceptual emphases of the course. It's also considerably harder than Bird et al. Make sure you are able to explain the most important concepts (*n*-gram, TF-IDF, LSA), but don't worry about every last detail.
    3. **Jurafsky and Martin** is the most theoretical of the three, but its treatment of word embeddings, sentiment lexicons, and word sense disambiguation is invaluable, and because it's more conceptual and our exam will not include practical programming exercises, it is important that you be able to discuss the main concepts here explained.
    4. The two blog posts by **Lynn** (only one of which was mandatory reading) offer both a more accessible introduction to word embeddings and more hands-on practice than what our textbooks can offer. You'll want to read them well and come prepared to discuss the concept, workings, and method of word embeddings at some length.
    5. I gave a choice of reading between **Tiedemann** and **Jurafsky ch. 13**, so I can't expect you to have studied them except insofar as they are explained in the slides. A broad understanding of word alignment may be useful, but don't bother memorizing which IBM model does what.
    6. **Searle** was more of an introduction to the meaning of meaning in the context of computing, but you should nevertheless be able to discuss the main points (the Chinese room argument, strong AI, weak AI).
    7. The **Langeslag** video is important inasmuch as we spend a week or two on the exact same topic, but if you followed class discussion the video should contain little extra information.
    8. Be able to discuss the main findings of **Ruder et al.** in general outline (i.e. in no more than a paragraph).
    9. **Smith** is a useful overview and proof of concept of crosslingual alignment of word embeddings.

10. The **Langeslag** introduction to CLTK is somewhat helpful as an overview of features.
11. The remaining sources (**Johnson**, **Wunderlich et al.**, **Torabi**) are of lesser importance.

- **Jupyter notebooks** can do a lot to help you understand practical implementation, but they won't be much help with the exam, which will be more theoretically oriented.
- **Student presentations** will play no role on the exam.

Record of Programming Tasks

We have tried our hands at the following methods either as homework or in class:

- Lexical diversity analysis
- Lemmatization
- Cosine similarity/dot product
- Classification with a naive Bayes supervised learning algorithm (gendered names/word senses)
- Document topic vectors
- Word embedding/word vectors
- Sentiment analysis
- Word alignment

Whereas you won't have to repeat the programming work on the exam, you'll want to be able to explain what steps are involved in carrying out these tasks, and discuss them intelligently.