Homework 3: Word embeddings

Deadline:

To be decided

Homework 3a understanding of word embedding

1. Train embeddings on 2 books from gutenberg, save the models on disk
2. Use a few input searches with .most_similiary() of gensim
3. Visualize the two models
4. For one model, evaluate the result of a sample query with the P@k (P@5) for the example query
5. Try different settings, different window sizes (2,5,10) and see how the evaluation measure changes
6. Compare the results of SkipGram versus CBOW algorithms, again with P@5
7. Make sentence representations which create sentence vectors, and use idf weighting to weight the words in the sentence
8. Test a few example queries, and output the closest sentences to the example queries

---

~~Homework 4 of word embeddings:~~

~~**Goal:** Similar to what we discussed and did in class within "Use Case 1", you should select a book (series) of your choice (which you know), and **create test data for the "analogies" and "doesnt_match" tasks**, and then **run the evaluation** of the word-embedding model.~~
~~You can reuse the existing code from https://github.com/gwohlgen/nlp4is_word_embeddings~~
~~So the main task is to a) train the model, b) create the testing data, c) evaluate the model.~~

~~If you want to work on the "A song of Ice and Fire" or "Harry Potter" bookseries, then I (Gerhard Wohlgenannt) will provide you with the a different git repo[1], and you just have to do tasks b) and c).~~

~~**The steps are as follows:**~~

1. ~~**Select the text** (for example books or book series, or any other text corpus) you want to work on. The corpus can be in any language (English, Russian, …)~~
2. ~~**Clone** the code at https://github.com/gwohlgen/nlp4is_word_embeddings~~
3. ~~**Train** your model, for example with gensim, see the slides of unit3~~
4. ~~*(If you work with the ASOIF or Harry Potter books, skip 1-3 and use repo in footnote 1)*~~
5. ~~Create your **test tasks** for the **two task types** (analogy, doesnt_match)~~
6. ~~Use create_questions.py to create the **evaluation questions**~~
7. ~~Run the **evaluations** (before: update config.py to use your data)~~
8. ~~Look at the **results**, present them in a nice way!~~
9. ~~Think about **preprocessing** your corpus, eg. removing punctuation, etc~~
    a. ~~Do preprocessing (lemmatization)~~
    b. ~~Retrain the model~~
    c. ~~Re-run the evaluation → are there **any changes** in results?~~

[1] https://github.com/gwohlgen/digitalhumanities_dataset_and_eval