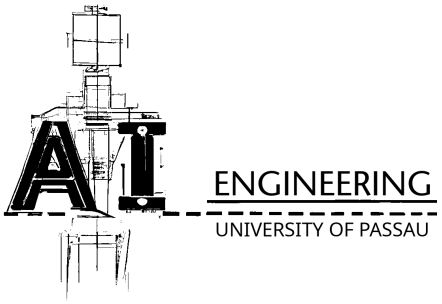
Deep Learning for Natural Language and Code

Exercise 1

Prof. Dr. Steffen Herbold
SoSe 2025
Due on 2025/05/08



General information for all exercises (read carefully!)

Within the "Deep Learning for Natural Language and Code Exercise," you will execute different tasks that relate various NLP concepts. The main goal of these exercises is to teach you how to develop approaches. Once you have gained this knowledge, you will have the opportunity to use your own solution and compare it with existing solutions from popular libraries. This means that these exercises are not just about knowing how to use libraries.

Problem description

One task that you need in NLP is to manipulate text. In this exercise, we will focus on converting free-form text into a Bag of Words (BoW) representation. For this task, we will use a dataset consisting of movie reviews.

- `https://ai.stanford.edu/~amaas/data/sentiment/`

In addition to the text manipulation task, you will build a model to classify between positive and negative reviews, and you will also develop a model for text generation.
In StudIP, a Jupyter Notebook is provided as a template for this exercise. In the template, a series of sections are provided in order to help you organize and structure the code. Additionally, some blocks of codes are also provided to facilitate some exercise tasks.

Data set description

For this exercise, we are going to work with the *Large Movie Review Dataset* [1]. This dataset was built for binary sentiment classification. It is composed of 25,000 highly polar movie reviews for training, and 25,000 for testing. Meaning that the middle values (i.e., scores of 5) are ignored. The dataset also contains unlabeled data. However, for this exercise, those are not going to be taken into account.

Programming tasks

Restrictions and tips for the tasks

The tasks must be solved using only standard Python libraries (i.e., no external libraries such as Scikit-Learn, NLTK, etc). The only exceptions where external libraries are allowed are:

- reading and writing LIBSVM BOW files,

- stemming (Porter Stemmer), and
- training the Random Forest model.

In all other parts of the implementation, only standard libraries may be used.
You could use Google Colab for executing the exercises (`https://colab.research.google.com`).

Classification and generation

1. Analyze the LIBSVM BOW that is included in the dataset and try to understand what type of preprocessing was done to the raw reviews in order to achieve that representation. For this, also analyze the code that is provided in the Jupyter.
2. Build a bag of words with the training data. For this, you have to:
   (a) Tokenize on spaces and punctuation.
   (b) Convert everything to lower case.
   (c) Remove punctuation.
   (d) Remove the terms that appear more often than a certain percent. You should be able to try with different values, e.g., 1%, 5%, 10%.
   (e) Use Porter stemmer for stemming.
3. Compare both BOW representations and make a hypothesis about the possible causes of the differences.
4. Use the BOW built in the second step as input and train a Random Forest to classify reviews as positive or negative sentiments.
5. Implement a simple Markov chain to estimate the probability of the new token, through counts, using the pipeline built in step two, but without the Porter stemmer (2.e).

Theoretical questions

1. Define to which categories (i.e., NLG, NLU, and NLP) the tasks 2, 4, and 5 belong.
2. Discuss drawbacks and advantages of this text processing pipeline (e.g., stemming, removing punctuation).
3. Review the following concepts:
   - Polysemy
   - Zeugma
   - Homonyms
   - Homographs
   - Homophones

   Once each concept has been reviewed, answer: 1) are all of those concepts important in NLP? Why? 2) What are the implications of those concepts when working with a BOW representation of texts?
4. Think about the performance of the Markov model, is it good? Why or why not?

References

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the*

49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.