Advanced Methods in Text Mining
Summer Semester 2024
Assignment 1 – NLP Basics

Instructor: Prof. Rafet Sifa, Dr. Lorrenz Sparrenberg,
Armin Berger, Maren Pielka, Tobias Deußer
University of Bonn

Introduction

In this assignment, we will explore some basic concepts of Natural Language Processing (NLP). This assignment will have 4 main parts totalling 100 points (pts.). The coding exercises will be implemented using the programming language Python 3.10 (or higher).

1. Preprocessing

In the lecture, we talked about various ways of preprocessing textual data. In here, we discuss different concepts and how they help the performance in later NLP tasks.

1.1 Concepts (5 Pts.)

Explain what the following text preprocessing concepts are, how they function, and why they are useful.

1. Sentence tokenization
2. Word tokenization
3. Part-of-speech (POS) tagging
4. Lemmatization
5. Stop word removal

1.2 Implementation (5 Pts.)

Build your own NLP preprocessing routine (a function named `process_paragraph(paragraph)`) that takes a paragraph as input, and splits the paragraph into sentences, applies word tokenization and lemmatization on all words. Furthermore, it should remove stop words from the input sentence. The function should return a list containing the processed sentences. You can use any open source package you like for this task, as long as each step is well documented.

2. Algebra Basics (10 Pts.)

- Describe briefly the following concepts:
  - What is a vector?
  - What is a matrix?
  - What is the rank of a matrix?
  - What is outer product expansion?

  - What is cosine similarity?

- Describe in a few sentences the concept of the gradient of a vector or matrix
- Explain briefly a linear classifier

3. Logistic Regression

Suppose we have 6,481 E-Mails in our dataset, of which 1,108 are spam E-Mails and 5,373 are non-spam E-Mails. Furthermore, you have two additional binary features for each E-Mail. Table 1 shows the number of occurrences for the two labels and the three possible features a E-Mail can have in this setup, i.e. `Colleague` for an E-Mail sent by a colleague, `Contacted` for an E-Mail from someone outside the company but contacted previously, and `Neither` if the sender of the E-Mail is neither a colleague nor someone previously contacted.

`Spam`	`Colleague`	`Contacted`	`Neither`	Total
`True`	11	98	999	1108
`False`	1829	1568	1976	5373
Total	1840	1666	2975	6481

Table 1: Frequency Table of Spam E-Mails.

3.1 Model Formulation (5 Pts.)

Write down the logistic regression model formulation in detail for predicting `Spam` from the variables described in Table 1. Specifically make sure you have defined and described the coefficients and variables in the model. Use `Neither` as the base level.

Hint: Look at page 16 on the practise session slides *ML Basics* for how the formula for a logistic regression is defined.

3.2 Calculating the Coefficients (10 Pts.)

Derive the maximum likelihood estimates for the parameters of the model, using `Neither` as the base level. Why can you derive it as a closed-form solution?

3.3 Implementation (10 Pts. + 5 bonus pts.)

Implement this logistic regression with the data provided in Table 1 using `sklearn`. What is the difference between the estimated coefficients and the coefficients calculated above? Why does this difference occur?

Bonus for +5 pts.: Implement the gradient descent-based method introduced in the lecture. What values of the learning rate seem to work the best? How do the results compare to the ones obtained from `sklearn`.
Hint: A good starting point is probably the documentation of `sklearn`:
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

3.4 Confusion Matrix (5 Pts.)

Given the calculated model parameter from subsection 3.2, create a confusion matrix of the predicted results.
Hint: See the lecture slide "Classification: Evaluation Measures" of the "Lecture 2: Preliminaries II".

3.5 Metrics (5 Pts.)

Given the confusion matrix from above, calculate Accuracy, Recall, Precision, and F_1 score. How can each of the metrics be interpreted?

3.6 Trustworthiness (10 Pts.)

Given the model and the data as described in Table 1, would you trust such a model to classify E-Mails correctly into Spam and Not-Spam? Justify your answer and point out strengths and weaknesses as well as possible remedies of this approach.

4. Tf-idf (5 Pts.)

Research what the term tf-idf stands for and describe how it is used to vectorize text (cite your sources!). Do you think an approach using a tf-idf encoder is superior to the one described above?

5. Classifying movie reviews using textual features (30 Pts.)

Given the Large Movie Review dataset[1] introduced by Maas et al. 2011, build a full review prediction pipeline with the following parts:

1. Reading in the raw data.
2. Using the pre-defined split.
3. Applying preprocessing. You are encouraged to use the function you wrote in 1.2, but can also use functions from any Python package you like.
4. Encoding your textual data using a tf-idf encoder and a Bag-of-Words encoder.
5. Training a logistic regression model on predicting whether the input is `negative` or `positve` using only your training set.
6. Predicting on your hold-out test set and reporting your achieved accuracy for both encoding methods.
7. Compare your results to a k-nearest neighbour search with at least $k = 3$ neighbours and tf-idf encoder.

A note on the performance of your models: You will **not** be graded on the accuracy of your model, only on the soundness of your approach.

References

Maas, Andrew L. et al. (June 2011). "Learning Word Vectors for Sentiment Analysis". In: *Proc. ACL-HLT*, pp. 142–150.

[1] `http://ai.stanford.edu/~amaas/data/sentiment/` or `https://huggingface.co/datasets/stanfordnlp/imdb`