TU Dortmund University
Department of Statistics
Dr. J. Rieger
M. Sc. K.-R. Lange

Winter Semester 2024/25
11/26/2024

Exercise sheet 7

Natural Language Processing

Hand-in (voluntarily): 12/02/2024 until 10:00 a.m. via Moodle
Please submit a `.py`, `.ipynb`, `.R` or `.rmd` file!

Task 1

In moodle you will find the files `emotion_dataset.csv` and `seed_words.json`.

They contain a corpus of Tweets that contain labels for a emotion classification task (anger, joy, sadness and optimism) and a list of possible seed words for each emotion to initialize an LDA with. In a seeded LDA, we force the model to not find topics itself, but the topics we plant into its training using the seed words. Using words referring to certain emotions, we force our LDA to model emotions rather than the content/topics of the Tweets. We will analyze, how we can utilize an unsupervised or a seeded LDA for text classification.
As this is not a default task for topic modeling, we cannot use the popular implementation but have to refer back to niche implementations, which is common for more niche NLP-tasks. For R, you can use the "seededlda" package, which contains decent documentation in its help-pages. For Python, you will find the file `lda_model.py` in moodle, which is a condensed version of this GitHub repository. See the recommended functions below for more help.

Task 2

Preprocess the texts so that they are fit for an analysis. Argue the use the preprocessing steps you take for the given analysis.

Task 3

Train five LDAs with $K = 4$ topics on your texts. For Python, set `n_features=10000` and set a high `n_iter`-value. For R, you will need the `quanteda::dfm` function (see the documentation of `seededlda::textmodel_lda`).

Task 4

Train a seeded LDA with the seeds from Task 1 on your texts using the same parameters as in Task 3. Look at the model's top words. Did the seeding work?

Task 5

Calculate the most dominant topic for each document for the models from task 3 and 4. Compare the results with the true emotion labels for eacht text by using a confusion matrix. Which model does better? Is it good enough or do you think, we need to find a more method more suited to a classifcation task?

Recommended packages & functions

R: `jsonlite::fromJSON()`, `data.table::fread()`, `quanteda::dfm`, `quanteda::dfm_trim()`,

seededlda::textmodel_lda(), seededlda::textmodel_seededlda(), table()
Python: json.load(), pandas.read_csv(), lda_model.LDA_Model,
lda_model.LDA_Model.documents_to_topic_model(), lda_model.LDA_Model.display_topics(),
sklearn.metrics.confusion_matrix()