Getting Started with the Topic Modeling Case Study (hands-on)

Topic-Modeling-Case-Study-Local.zip

Topic-Modeling-Case-Study-WS.zip

Download one of these zip files. *Topic-Modeling-Case-Study-Local.zip* contains the data and the notebooks that you can open locally using a Jupyter server. Alternatively, *Topic-Modeling-Case-Study-WS.zip* is a zip archive file containing the files needed to complete this study case in Watson Studio. You can directly upload this last zip file when creating a new Watson Studio project.

You will need the following files to complete this case study (all can be found in the above .zip file)

- topic-modeling-case-study.ipynb
- movie_reviews dataset (as a .csv format in the WS version and as a set of .txt files in the Local version)

The topic_modeling-case-study notebook will provide code snippets and instructions for the exercises. We strongly encourage that you try the exercises on your own before referring to the solutions notebook provided at the end of the case study.

Guiding principle for the case study

Topic modeling is a commonly used form of dimensionality reduction. When we use visualization tools to explore the results of topic modeling one strong hope is that we can identify features that are relevant to the domain. These insights can be in turn be transformed into new features that then can be used directly or appended to the feature matrix. We use topic modeling in this case study to enables domain specific feature engineering.

We will use the benchmark dataset movie reviews dataset [1]. The case study will make use of the NLTK package and you will want to ensure that you have run the following either as part of your notebook or at some point before.

Note: IBM Watson Natural Language Understanding, Natural Language Classifier, and Watson Discovery make use of pre-trained models for many different use cases. They may also be custom trained for specific knowledge domains.

```
import nltk
nltk.download('all')
```

The visualization of topics in a large corpus can be challenging. We will use the software pyLDAvis that integrates directly into Jupyter notebooks. This is based on the LDAvis project [2]. To install it use:

```
pip install pyldavis
```

---

[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*, 142–150. Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/P11-1015.

[2]: Carson Sievert and Kenneth Shirley. Ldavis: a method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. 06 2014.
URL: https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf, doi:10.13140/2.1.1394.3043.