Topik Documentation

Release 0.1.0

Christine Doig

October 10, 2015

Contents

1 What's a topic model? 3

2 Yet Another Topic Modeling Library 5

3 Getting Started 7

4 Contents 9
- 4.1 User Guide . . . 9
- 4.2 Developer Guide . . . 10
- 4.3 Reference Guide . . . 10

5 Useful Topic Modeling Resources 11
- 5.1 Python libraries . . . 11
- 5.2 R libraries . . . 11
- 5.3 Other . . . 11
- 5.4 Papers . . . 11

6 License Agreement 13

7 Indices and tables 15

8 Footnotes 17

Topik is a Topic Modeling toolkit.

CHAPTER 1

What's a topic model?

The following three definitions are a good introduction to topic modeling:

- A topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents [1].
- Topic models are a suite of algorithms that uncover the hidden thematic structure in document collections. These algorithms help us develop new ways to search, browse and summarize large archives of texts [2].
- Topic models provide a simple way to analyze large volumes of unlabeled text. A "topic" consists of a cluster of words that frequently occur together [3].

[1] http://en.wikipedia.org/wiki/Topic_model.
[2] http://www.cs.princeton.edu/~blei/topicmodeling.html
[3] http://mallet.cs.umass.edu/topics.php

CHAPTER 2

Yet Another Topic Modeling Library

Some of you may be wondering why the world needs yet another topic modeling library. There are already great topic modeling libraries out there, see *Useful Topic Modeling Resources*. In fact *topik* is built on top of some of them.

The aim of *topik* is to provide a full suite and high-level interface for anyone interested in applying topic modeling. For that purpose, *topik* includes many utilities beyond statistical modeling algorithms and wraps all of its features into an easy callable function and a command line interface.

Topik's desired goals are the following:

- Provide a simple and full-featured pipeline, from text extraction to final results analysis and interactive visualizations.
- Integrate available topic modeling resources and features into one common interface, making it accessible to the beginner and/or non-technical user.
- Include pre-processing data wrappers into the pipeline.
- Provide useful analysis and visualizations on topic modeling results.
- Be an easy and beginner-friendly module to contribute to.

CHAPTER 3

Getting Started

To demonstrate the ease of a typical *topik* workflow, we'll provide two examples: using the command line interface and using the method `topik.run.run_topic_model`.

- Using the command line interface

To get help you can always type `topik --help`.

```
$ topik --help

Usage: topik [OPTIONS]

  Run topic modeling

Options:
  -d, --data TEXT         Path to input data for topic modeling  [required]
  -f, --format TEXT       Data format provided: json_stream, folder_files,
                          large_json  [required]
  -m, --model TEXT        Statistical topic model: lda_batch, lda_online
  -o, --output TEXT       Topic modeling output path
  -t, --tokenizer TEXT    Tokenize method to use: simple, collocations,
                          entities, mix
  -n, --ntopics INTEGER   Number of topics to find
  --prefix_value TEXT     In 'large json' files, the prefix_value to extract
                          text from
  --event_value TEXT      In 'large json' files the event_value to extract text
                          from
  --field TEXT            In 'json stream' files, the field to extract text
                          from
  --help                  Show this message and exit.
```

The following example runs the default model LDA(batch) over a json stream, extracting the field *text* with simple word tokenization.

```
$ topik -d ./topik/tests/data/test-data-1.json -f json_stream -o ./test -n 3 --field text -t entities
```

- Using `topik.run.run_topic_model`

The same previous example using `run_topic_model` would be:

```
>>> from topik.run import run_topic_model
>>> run_topic_model(data='./topik/tests/data/test-data-1.json', format='json_stream', n_topics=3, fie
                    dir_path='./topic_model')
```

To understand `topik`'s output and results interpretation, see `Topik Output`.

CHAPTER 4

Contents

4.1 User Guide

4.1.1 Installation

Topik is meant to be a high-level interface for many topic modeling utilities (tokenizers, algorithms, visualizations...), which can be written in different languages (Python, R, Java...). Therefore, the recommended and easiest way to install *Topik* with all its features is using the package manager *conda*. Conda is a cross-platform, language agnostic tool for managing packages and environments.

```
$ conda install -c memex topik
```

There is also the option of just installing the Python features with pip.

```
$ pip install topik
```

Warning: The pip installation option will not provide all the available features, e.g. the LDAvis R package will not be available.

Requirements

Topik's requirements are:

- gensim
- pattern
- textblob
- nltk
- pandas
- blaze
- bokeh
- numpy
- into

4.1.2 Introduction Tutorial

In this tutorial we will examine *topik* with a practical example: Topic Modeling for Movie Reviews.

- The Movie Review Dataset
- Using the high-level interface `run_topic_model`
- Creating your own custom topic modeling flow
- Analyzing the results

The Movie Review Dataset

In this tutorial we are going to use the Sentiment Polarity Dataset Version 2.0 from Bo Pang and Lillian Lee. This dataset is distributed with NLTK with permission from the authors.

You can download the individual dataset from NLTK, or download all of ntlk's dataset, running the following commands from the python interpreter:

For more information on the datasets and download options visit NLTK data.

Instead of using the dataset in for *sentiment analysis*, its initial purpose, we'll perform *topic modeling* on the movie reviews. For that reason, we'll merge both folders *pos* and *neg*, to one named *reviews*.

High-level interfaces

As mentioned in the introduction page, there a two high-level interfaces: the command-line interface and the function topik.run()

Custom topic modeling flow

Analyzing the results

4.2 Developer Guide

4.3 Reference Guide

CHAPTER 5

Useful Topic Modeling Resources

- Topic modeling, David M. Blei

5.1 Python libraries

- Gensim
- Pattern
- TextBlob
- NLTK

5.2 R libraries

- lda
- LDAvis

5.3 Other

- Ditop

5.4 Papers

- Probabilistic Topic Models by David M.Blei

CHAPTER 6

License Agreement

topik is distributed under the BSD 3-Clause license.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

CHAPTER 8

Footnotes