News text visualization

based on LDA model algorithm implementation

Yuchen Tan 20032211

CONTENTS

Project Description

1.1 Background

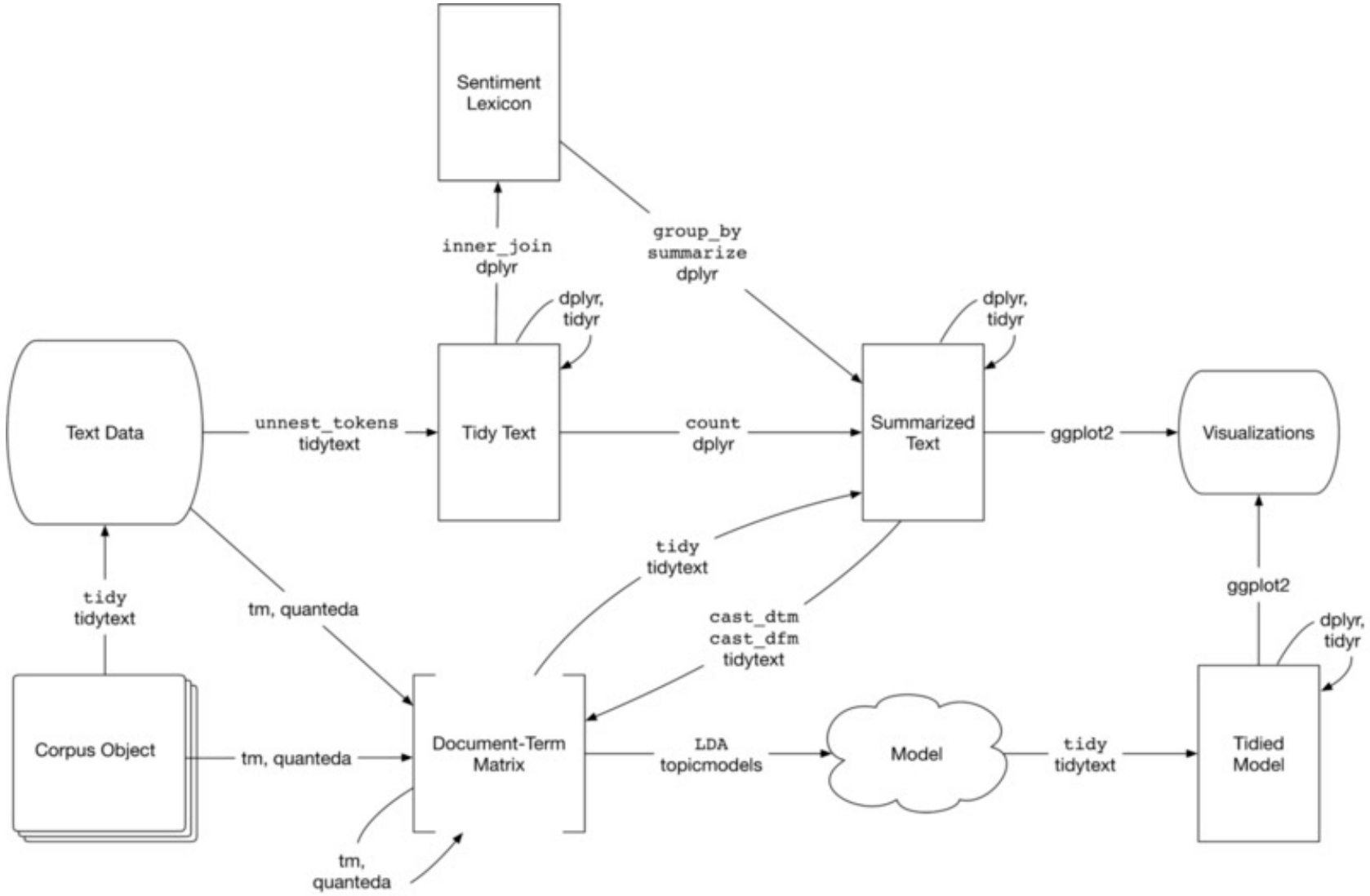
LDA topic model was first proposed by David M. Blei, Andrew Y. Ng and Michael I. Jordan in 2002. In recent years, with the rise of social media, textual data has become an increasingly important source of analysis; the huge amount of textual data has put forward new demands on the analytical ability of social science researchers, so LDA As a probabilistic model that can extract topics from a large amount of text, topic models are increasingly used in social science research such as topic discovery and document tagging.

With the development of information technology, people are living in a world full of information. While enjoying the convenience brought by various information services, we also have to face the situation of too much information that is difficult to handle.

1.2 Preparation

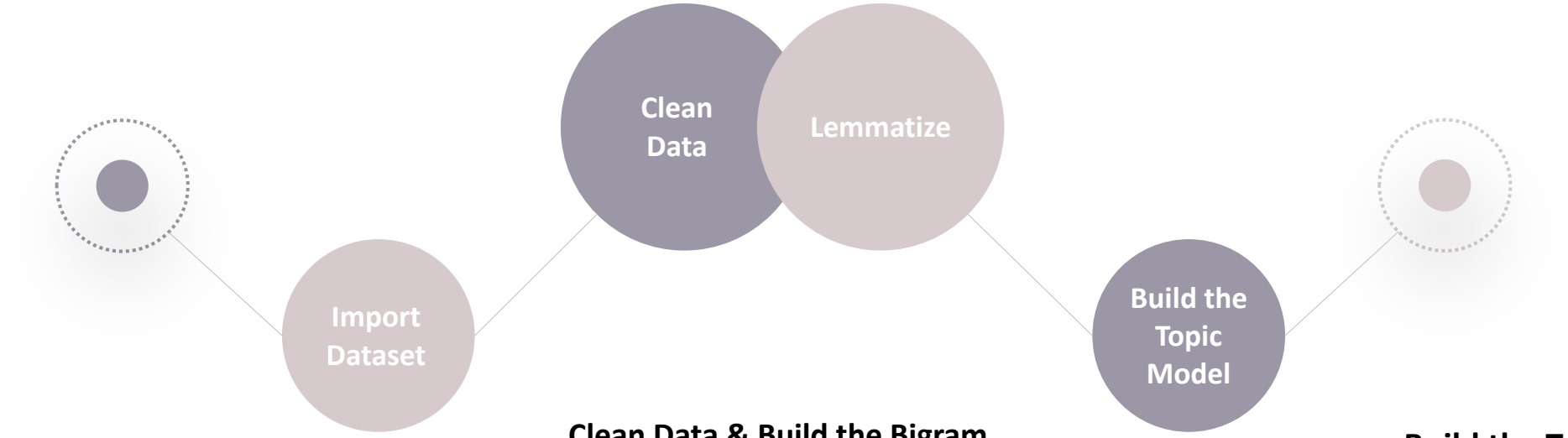
1.2.1 LDA Introduction

LDA Introduction



Data Collection

Please enter the title Please enter the title



Import News Groups Dataset

I first imported the newsgroup dataset and kept only 4 target_names categories. These four categories are: motorcycle news, sports news, political news, and religious news.

Clean Data & Build the Bigram, Trigram Models and Lemmatize

Removing the emails, new line characters, single quotes and finally split the sentence into a list of words. Next, lemmatize each word to its root form, keeping only nouns, adjectives, verbs and adverbs.

Build the Topic Model

The decimal after each word can be considered as the probability that the word belongs to the topic, and the probability sum of all words under the topic is 1.

Build the Topic Model

```
[(0,
  '0.012*"state" + 0.012*"israeli" + 0.011*"people" + 0.011*"kill" + '
  '0.009*"attack" + 0.009*"government" + 0.008*"war" + 0.007*"turkish" + '
  '0.006*"soldier" + 0.006*"greek"'),
 (1,
  '0.020*"game" + 0.018*"bike" + 0.017*"write" + 0.012*"article" + '
  '0.009*"rider" + 0.008*"list" + 0.008*"ride" + 0.007*"score" + '
  '0.006*"motorcycle" + 0.006*"helmet"'),
 (2,
  '0.017*"team" + 0.015*"year" + 0.012*"time" + 0.011*"write" + 0.009*"well" + '
  '0.009*"first" + 0.009*"play" + 0.008*"look" + 0.008*"help" + 0.008*"name"'),
 (3,
  '0.014*"people" + 0.012*"write" + 0.010*"believe" + 0.008*"reason" + '
  '0.007*"evidence" + 0.006*"question" + 0.006*"thing" + 0.006*"article" + '
  '0.006*"claim" + 0.005*"faith"')]
```

01

To build the LDA topic model using LdaModel(), we need the corpus and the dictionary. Let's create them first and then build the model. The trained topics (keywords and weights) are printed below as well.

02

If we examine the topic key words, they are nicely segregate and collectively represent the topics we initially chose: Christianity, Hockey, MidEast and Motorcycles.

Analyze the Text

3.1 Dominant topic

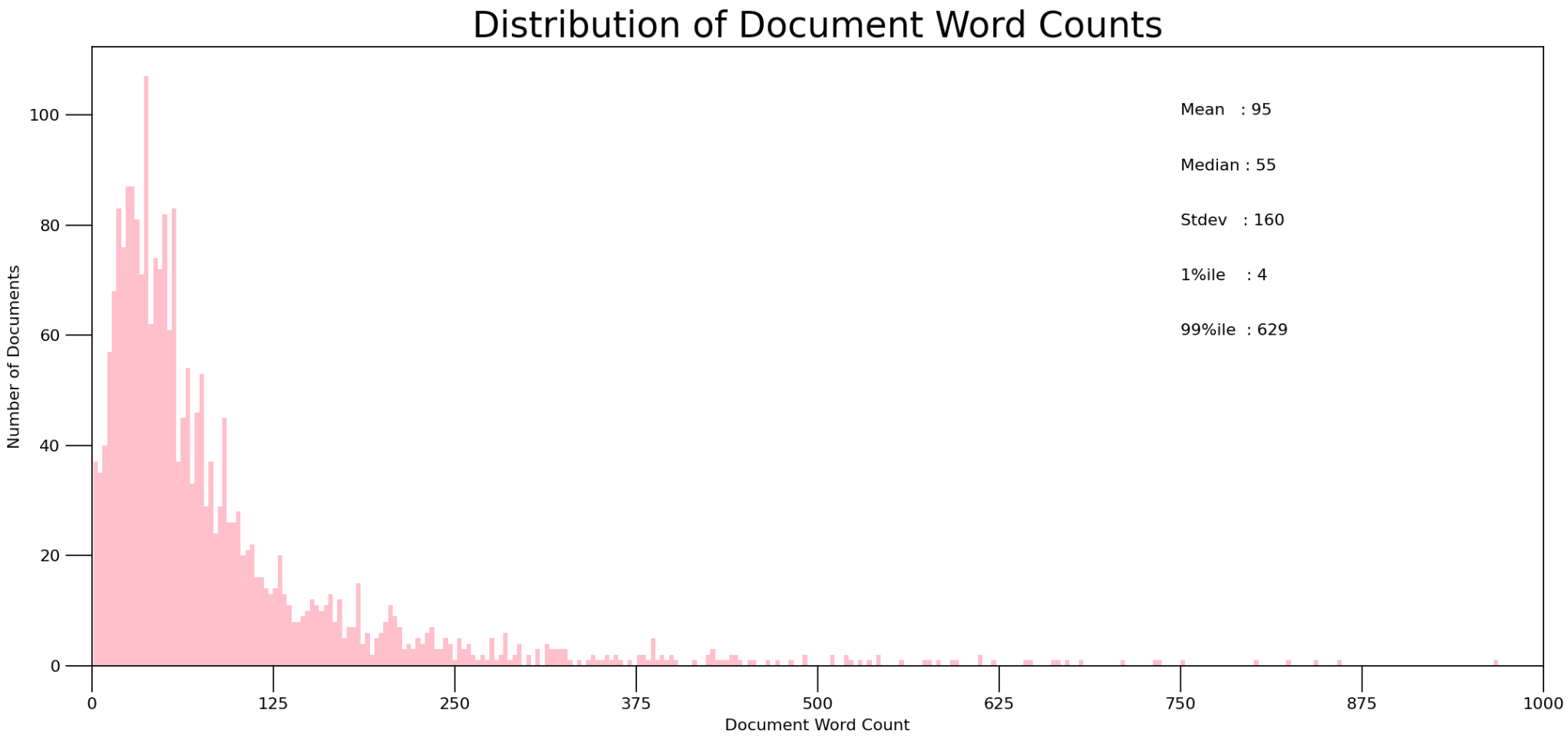
	Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
0	0	1.0	0.8037	game, bike, write, article, rider, list, ride,...	[summary, worth, expire, keyword, ducati, gts,...
1	1	2.0	0.8796	team, year, time, write, well, first, play, lo...	[group, article, write, course, time, prior, t...
2	2	3.0	0.7593	people, write, believe, reason, evidence, ques...	[write, new, group, maybe, cover, already, cur...
3	3	0.0	0.5978	state, israeli, people, kill, attack, governme...	[article, write, article, write, article, writ...
4	4	2.0	0.8912	team, year, time, write, well, first, play, lo...	[goalie_mask, organization, ist, article, writ...
5	5	1.0	0.6098	game, bike, write, article, rider, list, ride,...	[bmw_moa_member, read, internet, access, syste...
6	6	2.0	0.5123	team, year, time, write, well, first, play, lo...	[require, organization, article, write, articl...
7	7	1.0	0.4970	game, bike, write, article, rider, list, ride,...	[looking, movie, bike, university, latech, sum...
8	8	3.0	0.8834	people, write, believe, reason, evidence, ques...	[organization, university, follow, thread, tal...
9	9	0.0	0.6453	state, israeli, people, kill, attack, governme...	[day, night, round, male, inhabitant, article,...

	Topic_Num	Topic_Perc_Contrib	Keywords	Representative Text
0	0.0	0.9667	state, israeli, people, kill, attack, government, war, turkish, soldier, greek	[armenian, genocide, people, article, write, ranada, ermeniler, boyle, icin, bulunan, oldurmuler...
1	1.0	0.9125	game, bike, write, article, rider, list, ride, score, motorcycle, helmet	[hockey, draft, price, list, price, list, week, buy, team, player, pit, det, tor, pit, pit, tor,...
2	2.0	0.9783	team, year, time, write, well, first, play, look, help, name	[article, cire, write, write, write, captain, ever, traded_resigne, stripe, title, season, appre...
3	3.0	0.9871	people, write, believe, reason, evidence, question, thing, article, claim, faith	[article, write, affirm, absolute, scripture, believe, truth, reveal, truth, word, therefore, ho...

3.2 Representative sentence

Data Visualization

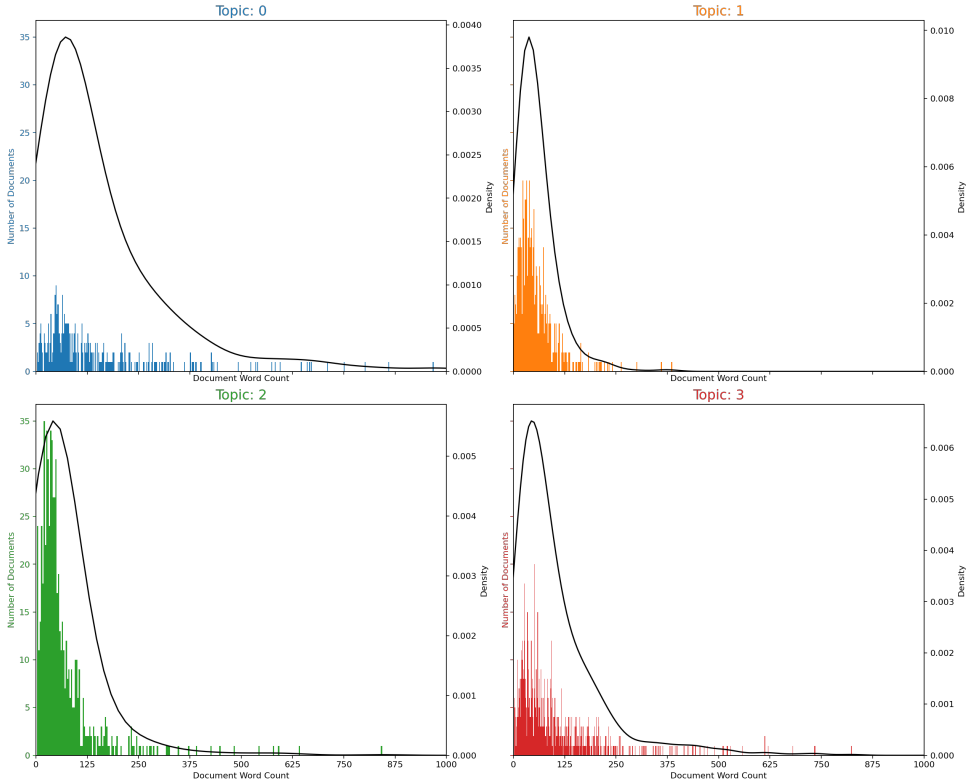
When working with a large number of documents, we want to know how big the documents are as a whole and by topic. I also calculate mean, median and stedv of it.



Data Visualization

Here’s the Distribution of Document Word Counts by Dominant Topic . Though I’ve already seen what are the topic keywords in each topic, a word cloud with the size of the words proportional to the weight is a pleasant sight.

Distribution of Document Word Counts by Dominant Topic



Word Cloud

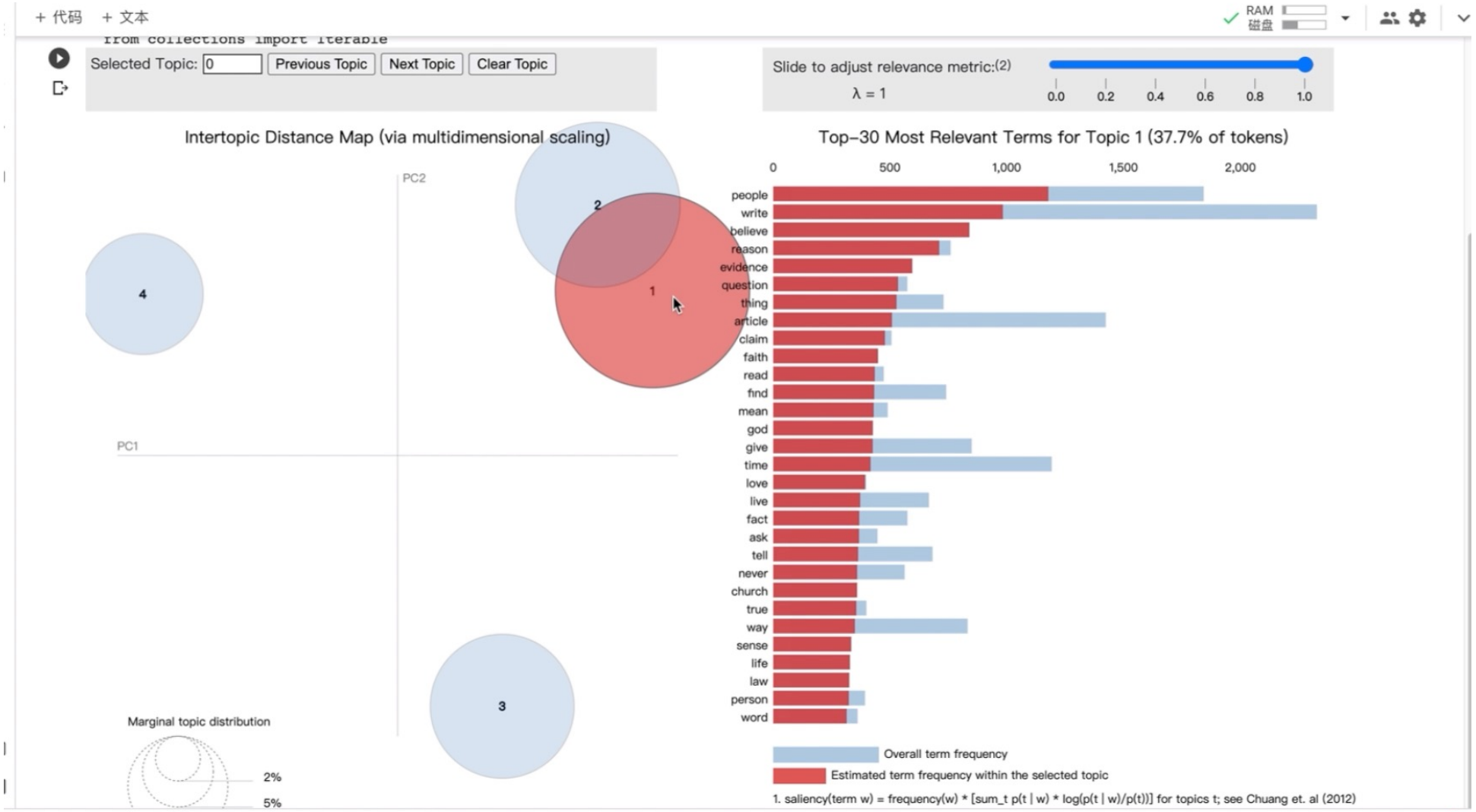
t-SNE Clustering Chart

■ t-SNE Clustering Chart

t-SNE (t-Distributed Stochastic Neighbor Embedding) is a very popular nonlinear dimensionality reduction technique, mainly used to visualize high-dimensional data.

The main purpose of using t-SNE here is to visualize the structure of the data, such as how many clusters the data are roughly clustered into, which clusters are closer together, and so on.

pyLDAvis


+ 代码 + 文本
RAM
磁盘
from collections import Iterable
Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
Marginal topic distribution
2%
5%
Top-30 Most Relevant Terms for Topic 1 (37.7% of tokens)
0
500
1,000
1,500
2,000
people
write
believe
reason
evidence
question
thing
article
claim
faith
read
find
mean
god
give
time
love
live
fact
ask
tell
never
church
true
way
sense
life
law
person
word
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)

Conclusion

A

I started from scratch by importing, cleaning and processing the newsgroups dataset to build the LDA model. Then we saw multiple ways to visualize the outputs of topic models including the word clouds and sentence coloring, which intuitively tells you what topic is dominant in each topic. A t-SNE clustering and the pyLDAVis are provide more details into the clustering of the topics.

B

It was an interesting attempt to help me better understand the LDA model and the flexibility of the various visualizations. At the same time, I found that there are also many people from other countries who use their own languages to build LDA topic models, such as Chinese. This is different from the way text is processed in English in many ways. I think in the future I can try theme models in multiple languages and use sentiment analysis to complete more visual charts and do some more complex representations.

Thank you!