Topic Modeling on Reddit with spaCy, Latent Dirichlet Allocation and Gensim

Project for

Social Data Analytics and

Text Analytics and Natural Language Processing

Leonardo Dalle Luche, 601220

Master in Data Analytics, 2022/2023

Roma Tre University

1. Introduction

The following project has been made with the intent of modeling and discovering the main topics within the data science community on Reddit (a social network similar to Twitter). Reddit offers free API to extract its subreddits. A subreddit is basically the name of a community, and a community contains many posts. The following snippets show the result of an API call given four parameters.

```
# Call the "top" last month's 100 posts within the "datascience" subreddit
subreddit = 'datascience'
type_of_posts = 'top'          # controversial, best, hot, new, random, rising, top
limit = 100
timeframe = 'month'            #hour, day, week, month, year, all
```

	title	url	comments
0	What opinion about data science would you defe...	https://i.redd.it/20r6sbok4a4c1.jpg	629
1	A gentle reminder that the market is a shitsho...	https://www.reddit.com/r/datascience/comments/...	179
2	125k offer as a data scientist but I have no i...	https://www.reddit.com/r/datascience/comments/...	277
3	6 months as a Data Science freelancer	https://www.reddit.com/r/datascience/comments/...	117
4	Every AI startup right now	https://i.redd.it/t08yextu6x2c1.png	28
...	...	...	...
95	Job advice, dealing with higher ups	https://www.reddit.com/r/datascience/comments/...	30
96	Handed a dataset, what's your sniff test?	https://www.reddit.com/r/datascience/comments/...	24
97	MS Statistics vs. MS CS	https://www.reddit.com/r/datascience/comments/...	37
98	Companies with good work-life balance reputati...	https://www.reddit.com/r/datascience/comments/...	26
99	Anyone left corporate to go the entrepreneur r...	https://www.reddit.com/r/datascience/comments/...	24

100 rows × 3 columns

To simplify the analysis, only the top one hundred posts within a timeframe of one month have been considered. Moreover, the “title” column of the DataFrame has been used for topic modeling.

2. Latent Dirichlet Allocation (LDA)

2.1 Pre-Processing

In order to use the LDA model to perform topic modeling on Reddit posts, a corpus and a Gensim dictionary are required. To obtain such data structures, the following steps have been taken into consideration:

- Tokenization of the text (the process of converting a sequence of text into smaller parts, known as tokens)

```
# Tokenize and preprocess the titles of Reddit posts using Spacy
tokenized_docs = []

# Append to the list 'tokenized_docs' only if the token is not a stop word (not is_stop) and consist of alphabetic characters (is_alpha)
for doc in nlp.pipe(df['title']):
    tokenized_docs.append( [token.lemma_.lower() for token in doc if (token.is_alpha) and (not token.is_stop) ] )
```

- Creation of the dictionary (a mapping between words and their ids)

```
# Create a Gensim dictionary and corpus
dictionary = corpora.Dictionary(tokenized_docs)
```

- Creation of a corpus of bag of words (a model of text which uses a representation of text that is based on an unordered collection (or "bag") of words.)

```
# Create the corpus for LDA model, from doc to bag of words (bow)
corpus = [dictionary.doc2bow(doc) for doc in tokenized_docs]
```

2.2 LDA

To perform topic modeling using the LDA model, the following parameters have been used:

```
# Define the number of topics (k) to discover and the number of iteration
n_topics = 5
n_iteration = 30
```

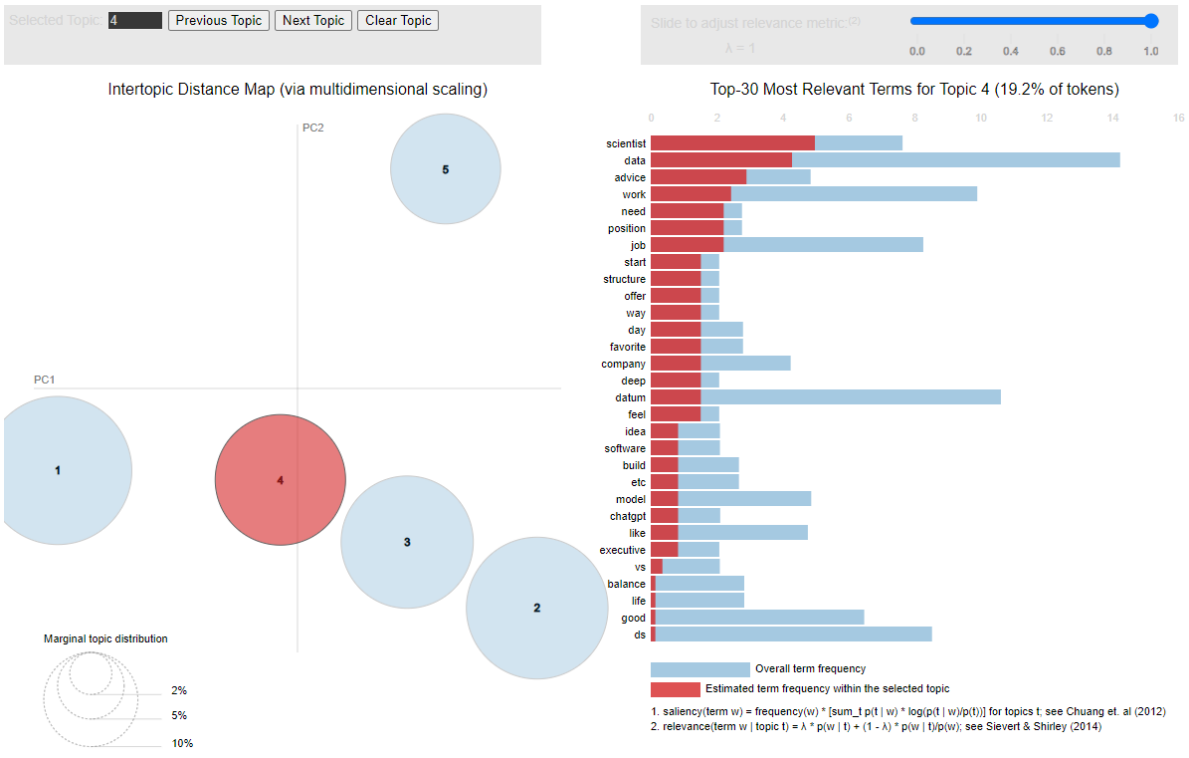
This means that the LDA model will extract five topics given the pre-processed text, in a number of thirty iterations during training. Please note that alpha and eta parameters have been set to ‘auto’ to get better performance from the model.

```
# Build the LDA (Latent Dirichlet Allocation) model
lda_model = models.LdaModel(corpus, num_topics=n_topics, id2word=dictionary, passes=n_iteration, alpha='auto', eta='auto')
```

```
# Print the topics
for idx, topic in lda_model.print_topics(num_words=5):
    print(f"Topic {idx}: {topic}")
```

```
Topic 0: 0.145*"data" + 0.130*"science" + 0.090*"good" + 0.068*"datum" + 0.046*"industry"
Topic 1: 0.121*"ds" + 0.117*"work" + 0.050*"year" + 0.038*"model" + 0.038*"life"
Topic 2: 0.055*"job" + 0.055*"datum" + 0.042*"cs" + 0.042*"ms" + 0.042*"role"
Topic 3: 0.097*"scientist" + 0.084*"data" + 0.057*"advice" + 0.048*"work" + 0.043*"need"
Topic 4: 0.057*"think" + 0.057*"recruiter" + 0.057*"like" + 0.039*"etc" + 0.039*"build"
```

From the output we can see the distribution of probabilities assigned to each word within a topic by the model. Thanks to pyLDAvis we can print out an interactive plot of LDA model output:



From the plot, we can see that the topic 4 is selected. On the right side, we can see the top 30 most relevant terms for topic 4. It's interesting to note that the top-4 words are "scientist", "data", "advice", and "work", suggesting that users probably asked for job advice related to data scientist positions within the community.

3. WordCloud

3.1 Common words between topics

From this snippet we can see common words between all the topics, with a probability threshold greater or equal than 0.04. It's interesting to see that the words "datum", "data", and "work" are common to some topics.

```
# Define a threshold to filter out words with probability < threshold
threshold = 0.04

for i in range(len(result)):
  j = i + 1
  while j < len(result):
      # Call utility function "get_common_words_between_topics" passing topic_x and topic_y
      # Get common words between two topics
      common_words = get_common_words_between_topics(result[i], result[j], threshold) # result[i] = topic_x, result[j] = topic_y

      if len(common_words) > 0:
        for common_word in common_words:
          print(f'Common word between topic-{i} and topic-{j}: {common_word}')

      j = j + 1
```

```
Common word between topic-0 and topic-2: datum
Common word between topic-0 and topic-3: data
Common word between topic-1 and topic-3: work
```

3.2 Plot of WordCloud

After concatenating all words and probabilities into a single dictionary, from the following plot we can see the overall words with the highest probability. As excepted, the words "science", "scientist", "ds" (which is an internet slang standing for "data science"), "data", etc. are all related to data science.

References

spaCy: https://spacy.io/api/token

Gensim and LDA model: https://radimrehurek.com/gensim/models/ldamodel.html

WordCloud: https://amueller.github.io/word_cloud/