Topic modeling for gender stereotypes

Department of Information Systems
University of Maryland Baltimore County

V.K.N Kamlesh Pai
(ob29417@umbc.edu)

Pranali Kulkarni
(xo66738@umbc.edu)

Abstract:

In machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. Topic modeling is a frequently used text-mining tool for discovery of hidden semantic structures in a text body.A **word embedding** is a learned representation for text where **words** that have the same meaning have a similar representation. It is this approach to representing **words** and documents that may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems. In this paper geometric relationship is used to capture a meaningful semantic relationship between the corresponding words. Thus, this embedding helps to capture societal shifts eg: how certain adjectives become associated with certain populations over time. Thus, it opens a fruitful intersection between machine learning and quantitative social science.

In this paper we use topic modelling technique to find the top 10 words in the CORPUS and thus find the top 3 topics of each word.

Dataset:

In this paper we learn and use two different types of dataset. i.e. COHA and COCA.

The Corpus of Contemporary American English (COCA) is the only large, genre-balanced corpus of American English. COCA is probably the most widely-used corpus of English, and it is related to many other corpora of English that we have created, which offer unparalleled insight into variation in English. The corpus contains more than one billion words of text (20 million words each year 1990-2019) from eight genres: spoken, fiction, popular magazines, newspapers, academic texts, and (with the update in March 2020): TV and Movies subtitles, blogs, and other web pages.

The Corpus of Historical American English (COHA) is the largest structured corpus of historical English. It is related to many other corpora of English that we have created, which offer unparalleled insight into variation in English. If you are interested in historical corpora, you might also look at our Google Books (see comparison), Hansard, and TIME corpora. COHA contains more than 400 million words of text from the 1810s-2000s (which makes it 50-100 times as large as other comparable historical corpora of English) and the corpus is balanced by genre decade by decade. The creation of the corpus results from a grant from the National Endowment for the Humanities (NEH) from 2008-2010.

Motivation:

It is obvious that every mathematical system or algorithm needs some sort of numeric input to work with. However, while images and audio naturally come in the form of rich, high dimensional vectors (i.e. pixel intensity for images and power spectral density coefficients for audio data), words are treated as discrete atomic symbols.

The naive way of converting words to vectors might assign each word a one-hot vector in vocabulary size. This vector will be all zeros except one unique index for each word. Representing words in this way leads to substantial data sparsity and usually means that we may need more data in order to successfully train statistical models.

What mentioned above raises the need for continuous, vector space representations of words that contain data that can be leveraged by models. To be more specific we want semantically similar words to be mapped to nearby points, thus making the representation carry useful information about the word actual meaning.

Related Work:

1. Topic Modelling and word embedding

The Gaussian LDA model of Das et al. (2015) improves the performance of topic modeling by leveraging the semantic information encoded in word embeddings. Gaussian LDA modifies the generative process of LDA such that each topic is assumed to generate the vectors via its own Gaussian distribution. Similarly, to our MMSG model, in Gaussian LDA each topic is encoded with a vector, in this case the mean of the Gaussian. It takes pre-trained word embeddings as input, rather than learning the embeddings from data within the same model and does not aim to perform word embedding.

The topical word embedding (TWE) models of Liu et al.(2015) reverse this, as they take LDA topic assignments of words as input and aim to use them to improve the resultant word embeddings. The authors propose three variants, each of which modifies the skip-gram training objective to use LDA topic assignments together with words. In the best performing variant, called TWE-1, a standard skip-gram word embedding model is trained independently with another skip-gram variant, which tries to predict context words given the input word's topic assignment. The skip-gram embedding, and the topic embeddings are concatenated to form the final embedding.

2. Multi-prototype embedding models.

Multi-model embedding models are another pertinent profession. These models address lexical equivocalness by allocating numerous vectors to each word type, each corresponding to an alternate significance of that word. Reisinger what's more, Mooney (2010) proposes to bunch the events of each word type, in light of highlights removed from its content. Embeddings are then learned for each group. Huang et al. (2012) apply a comparative methodology, yet they utilize beginning single-model word embeddings to give the highlights utilized for bunching. These grouping techniques have some re- similarity to our point model pre-grouping step, in spite of the fact that their grouping is applied inside cases of a given word type, as opposed to all-around over-all word types, as in our strategies. This results in models with a bigger number of vectors than words, while we intend to discover less vectors than words, to diminish the model's intricacy for little datasets. Or maybe than utilizing an off-the-rack grouping calculation and at that point applying a random installing model to its output, our methodology plans to perform model-based bunching inside a general joint model of point/bunch assignments what's more, word vectors. Maybe the most comparable model to our own in the writing is the probabilistic multi-model implanting model of Tian et al. (2014), who treat the model task of a word as a dormant variable, expected drawn from a blend over models for each word. The embeddings are at that point prepared utilizing EM. Our MMSG model can be comprehended as the blended enrollment variant of this model, in which the models (vectors) are shared over all word types, and each word type has its own blended enrollment extent over the common models. While a comparable EM calculation can be applied to the MMSG, the E-step is significantly more costly, as we commonly want a lot increasingly shared vectors (frequently in the thousands) than we would models per single word type (Tian et al. utilize ten in their analyses). We utilize the Metropolis-Hastings-Walker calculation with the point model reparameterization of our model so as to address this by productively pre-understanding the E-step.

3. Mixed membership modeling

Mixed membership modeling is a flexible alternative to traditional clustering, in which each data point is assigned to a single cluster. Instead, mixed membership models position that individual entities are associated with multiple underlying clusters, to differing degrees, as encoded by a mixed membership vector that sums to one across the clusters(Erosheva et al., 2004; Airoldi et al., 2014). These mixed membership proportions are generally used to model lower-level grouped data, such as the words inside a document.Each lower-level data point inside a group is assumed to be assigned to one of the shared, global clusters according to the group-level membership proportions. Thus, a mixed membership model consists of a mixture model for each group, which share common mixture component parameters,

Data Format:

The input is a single file, with one line per document, and words represented by zero-based dictionary indices.

```
import os
import string
import pandas as pd
import io
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

```
#Reading the dataset into a common list

os.chdir("C:\Pranali\Independent Study\Dataset\coca1")
commonlist=[]
for i in os.listdir():

    f = open(i,"r")
    text = f.read()

    #splitting on white spaces
    words = text.split()
    commonlist.append(words)

len(commonlist)

```

```
#replace punction with whitespace

commonlist_no_punct=[]

for sublist in commonlist:
    table = str.maketrans('', '', string.punctuation)
    no_punct = [w.translate(table) for w in sublist]
    commonlist_no_punct.append(no_punct)
```

```python
#remove whitespaces

commonlist_no_whitespace = []

for sublist in commonlist_no_punct:
    new_x = [elem for elem in sublist if elem.strip()]
    commonlist_no_whitespace.append(new_x)
```

```python
# convert to lower case

commonlist_lcase = []

for sublist in commonlist_no_whitespace:
    x = [word.lower() for word in sublist]
    commonlist_lcase .append(x)
```

```python
#removing stopwords

stop_words = set(stopwords.words('english'))
commonlist_no_stopwords = []

for sublist in commonlist_lcase:
    output = []
    for x in sublist:
        if x not in stop_words:
            output.append(x)
    commonlist_no_stopwords.append(output)
```

```python
# Get the word count of each word

w_count = dict()

for sublist in commonlist_no_stopwords:
    for word in sublist:
        if word in w_count:
            w_count[word] = w_count[word] + 1
        else:
            w_count[word] = 1
```

```python
#Eliminate the words with count less than 10

new_dict = {key : value for (key, value) in w_count.items() if value > 9}
```

```python
#converting new dictionary to list

new_word_list=list(new_dict.keys())
```

```python
#Mapping the words with their resprctive indices

list_after_mapping = []

for sublist in commonlist_lcase:
    elem = {k: i for i, k in enumerate(new_word_list)}
    Output = list(map(elem.get, sublist))
    list_after_mapping.append(Output)
```

```python
#Eliminating None values from the list

final_output = []

for sublist in list_after_mapping:
    output = []
    for x in sublist:
        if x != None :
            output.append(x)
    final_output.append(output)
```

```python
# Making the output compatible for MMSG topic model
# Eliminating brackets and commas

string1 = str(final_output)[1:-1]
string2 = str(string1)[1:-1]
string3 = ''.join(string2.split(','))
string4 = '\n'.join(string3.split('['))
Final_file = ''.join(string4.split(']'))

#'Final_file' compatible for MMSG topic model


#Saving Final file
with open("C:\Pranali\Independent Study\Final Output\input file for MMSG", "w") as output:
    output.write(str(Final_file))
```

Running the code:

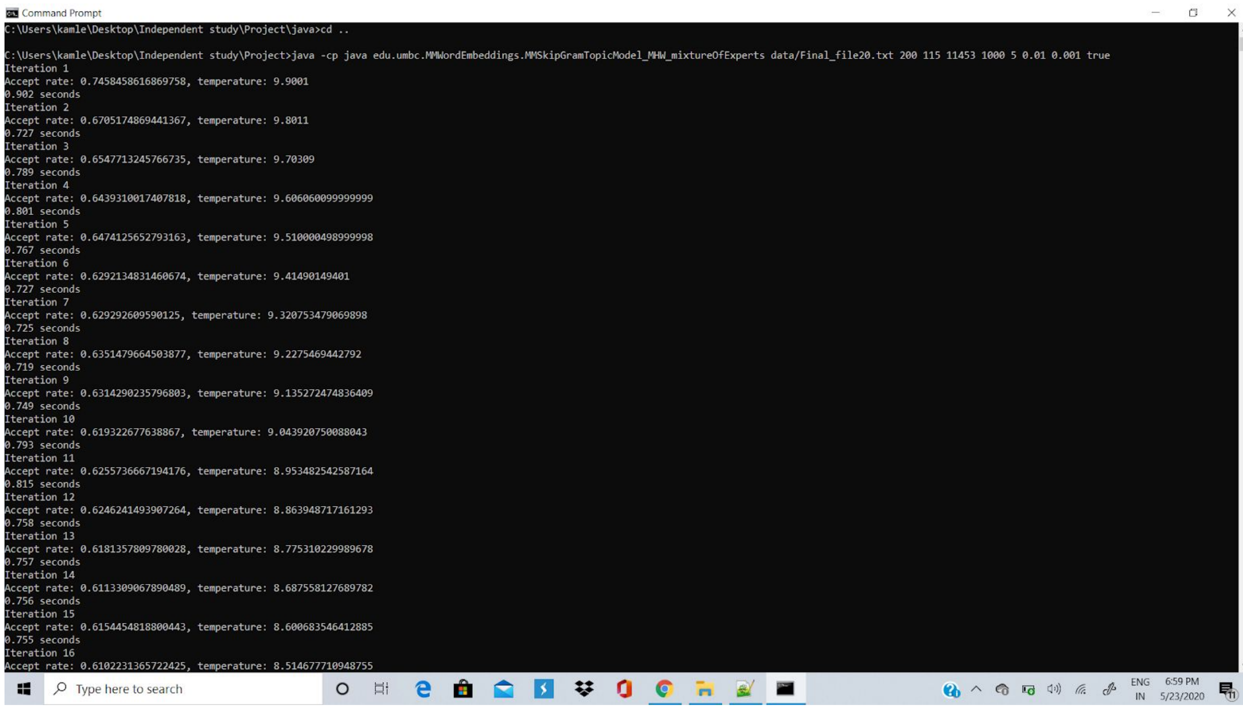
How we trained the model:

Input parameters given to train the model include number of topics, number of words, number of documents and number of iterations required.

Basic syntax for training a model in java is as follows:

```
java -cp java
edu.umbc.MMWordEmbeddings.MMSkipGramTopicModel_MHW_mixtureOfExperts filename
```

`numTopics numDocuments numWords numIterations contextSize alpha_k beta_w doAnnealing annealingFinalTemperature`



After running (it may take a while), this results in three files:

- MMskipGramTopicModel_topicAssignments.txt, in a format similar to the input data, but which contains topic assignments for each word
- MMskipGramTopicModel_wordTopicCountsForTopics.txt, which contains the count matrix for the topics (words by topics). Add the smoothing hyperparameter and normalize the columns to sum to one to obtain the topics' probability distributions over words
- MMskipGramTopicModel_wordTopicCountsForWords.txt, which contains the count matrix for the words' distributions over topics (words by topics). Add the smoothing hyperparameter and normalize the rows to sum to one to obtain the words' probability distributions over topics.

Getting Top 10 words in each topic:

```python
#Top 10 words
# Reading the output file generated from the MMSG Topic Model
df_topic=pd.read_csv("C:/Pranali/Independent Study/Final Output/MMSGTM_ForTopics.csv")
df_topic.set_index("WORD", inplace = True)


#Filter top 10 values for all columns (200 Topics)

new_df1 = df_topic.apply(lambda s, n: pd.Series(s.nlargest(n).index), axis=0, n=10)

#Saving the result in excel file
new_df1.to_excel (r'C:\Pranali\Independent Study\Final Output\top10_words.xlsx', index = True, header=True)
```

Preview of the output of Top 10 words in each topic:

```python
new_df1
```

3]:

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	...	Topic 191	Topic 192	Topic 193	Topic 194	Topic 195	T
0	p	p	narrator	like	announcer	p	p	nt	p	nt	...	mr	nt	nt	clemens	p	
1	nt	religious	barrie	new	rat	students	shelly	president	miners	p	...	p	one	children	conan	students	
2	would	liberal	1	nt	grandma	one	petey	would	health	people	...	said	voiceover	longing	pettitte	social	
3	one	freedom	2	voiceover	year	would	purple	know	coal	would	...	almonor	new	p	roger	studies	
4	us	academic	white	one	1	education	butterflies	going	http	one	...	would	like	know	said	student	
5	day	political	peter	idrive	cat	political	butterfly	voiceover	disease	could	...	one	p	like	mcnamee	sop	
6	said	religion	mrs	many	2	work	collection	kennedy	cwp	like	...	president	would	people	goldman	teachers	
7	back	one	mr	vanocur	emperor	may	wings	qwq	states	know	...	nt	vanocur	voiceover	truth	education	
8	time	human	sylvia	p	chinese	many	article	years	virginia	life	...	people	system	could	nt	tasks	
9	know	moral	play	time	ox	fact	copyright	time	workers	time	...	think	even	said	drugs	skills	

10 rows × 200 columns

Getting top 3 topics for each word:

```python
#Top 3 topics
# Reading the output file generated from the MMSG Topic Model
df_words=pd.read_csv("C:/Pranali/Independent Study/Final Output/MMSGTM_ForWords.csv")
df_words.set_index("WORD", inplace = True)


#Filter top 3 values for every row (11453 words)

new_df = df.apply(lambda s, n: pd.Series(s.nlargest(n).index), axis=1, n=3)

#Saving the result in excel file
new_df.to_excel (r'C:\Pranali\Independent Study\Final Output\top3_topics.xlsx', index = True, header=True)
```

Preview of the output of Top 3 topics for each word:

```
1 new_df
```

]:

WORD	0	1	2
section	Topic 195	Topic 101	Topic 25
issues	Topic 51	Topic 195	Topic 159
p	Topic 19	Topic 87	Topic 179
allan	Topic 130	Topic 77	Topic 96
bloom	Topic 130	Topic 31	Topic 19
...	...	...	...
jancrawford	Topic 118	Topic 200	Topic 42
ricktigner	Topic 58	Topic 1	Topic 2
bobsimon	Topic 167	Topic 66	Topic 32
tompapa	Topic 167	Topic 1	Topic 2
elizabethbernstei	Topic 167	Topic 1	Topic 2

11453 rows × 3 columns

Conclusion:

We have proposed a model-based technique for preparing interpretable corpus-explicit topic modeling for computational sociology, utilizing blended participation representations, Metropolis-Hastings-Walker inspecting, and NCE. Test results for expectation, regulated learning, and contextual analyses on condition of the Union advertisement dresses and COCA articles, show that top notch sheets and points can be gotten utilizing the technique. The outcomes feature the way that enormous information isn't always best, as area explicit information can be truly significant, in any event, when it is little. we intend to utilize this methodology for considerable sociology applications, and to address algorithmic predisposition and reasonableness issues.

We could draw some preliminary conclusions from our dataset. While the topics are a bit noisy, we do actually see some evidence of gender bias in the topics as shown in the figure below. The topics with the female names were mostly conversational and generic (people's names, common verbs, words relating to houses). One male name had a topic about philosophy, and the other male name had three topics about sports, both of which are stereotypically more associated with males than females.

WORD	0	1	2
mary	Topic 36	Topic 65	Topic 124
matthew	Topic 5	Topic 61	Topic 2

Mary			Matthew		
Topic 36	Topic 65	Topic 124	Topic 5	Topic 61	Topic 2
p	luke	p	announcer	voiceover	p
back	nt	jenna	rat	unidentifie	religious
one	sam	said	grandma	man	liberal
like	p	nt	year	officer	freedom
would	narrator	back	1	woman	academic
around	palazzo	door	cat	qwq	political
away	horty	room	2	2	religion
time	one	alison	emperor	rather	one
took	texas	like	chinese	1	human
thing	going	one	ox	right	moral

WORD	0	1	2
janet	Topic 112	Topic 67	Topic 37
johnson	Topic 179	Topic 19	Topic 148

janet			johnson		
Topic 112	Topic 67	Topic 37	Topic 179	Topic 19	Topic 148
voiceover	p	nt	p	p	p
qwq	said	going	said	said	olympic
sailor	room	think	game	team	said
unidentifie	one	people	nt	players	games
goldberg	would	get	team	nt	olympics
nt	back	know	last	one	world
1	nt	would	one	coach	team
msabrams	like	voiceover	like	player	medal
schlesinge	get	said	play	last	one
2	take	want	season	would	gold

Future Work:

We have only proceeded with topic modeling in this Corpus Data but we plan to work on word embeddings in the near future. Using the same dataset in future we can conclude and work on gender based biasing and analysis for eliminating the basic stereotypes.

References:

1. Hamilton DL, Trolier TK (1986) Stereotypes and Stereotyping: An Overview of the Cognitive Approach in Prejudice, Discrimination, and Racism (Academic, San Diego), pp 127–163.

2. Basow SA (1992) Gender: Stereotypes and Roles (Thomson Brooks/Cole Publishing Co, Belmont, CA), 3rd Ed.

3. Wetherell M, Potter J (1992) Mapping the Language of Racism: Discourse and the Legitimation of Exploitation (Columbia Univ Press, New York).

4. Holmes J, Meyerhoff M, eds (2004) The Handbook of Language and Gender(Blackwell Publishing Ltd, Oxford).

5. Coates J (2016) Women, Men and Language: A Sociolinguistic Account of Gender Differences in Language (Routledge, London).

6. Williams JE, Best DL (1977) Sex stereotypes and trait favorability on the adjective check list. Educ Psychol Meas 37:101–110.

7. Williams JE, Best DL (1990) Measuring Sex Stereotypes: A Multination Study (Sage Publications, Thousand Oaks, CA), Rev Ed.

8. Katz D, Braly K (1933) Racial stereotypes of one hundred college students. J Abnorm Soc Psychol 28:280–290.

9. Gilbert GM (1951) Stereotype persistence and change among college students. J Abnorm Soc Psychol 46:245–254.

10. Karlins M, Coffman TL, Walters G (1969) On the fading of social stereotypes: Studies in three generations of college students. J Pers Soc Psychol 13:1–16.

11. Devine PG, Elliot AJ (1995) Are racial stereotypes really fading? The Princeton trilogy revisited. Pers Soc Psychol Bull 21:1139–1150.

12. Diekman AB, Eagly AH (2000) Stereotypes as dynamic constructs: Women and men of the past, present, and future. Pers Soc Psychol Bull 26:1171–1188.

13. Bergsieker HB, Leslie LM, Constantine VS, Fiske ST (2012) Stereotyping by omission: Eliminate the negative, accentuate the positive. J Pers Soc Psychol 102:1214–1238.

14. Madon S, et al. (2001) Ethnic and national stereotypes: The Princeton trilogy revisited and revised. Pers Soc Psychol Bull 27:996–1010.