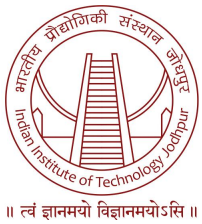


Natural Language Understanding - CSL7340

Statistical & Topic Analysis of Amazon Book Category Dataset

Ayanabha Ghosh (M21CS055)
Puja Gupta (M21MA004)
Shaonli Pal (M21MA007)

Colab File Link :
https://colab.research.google.com/drive/1oyQHAouCJzJeegcQ-es-_JHKF2OKoP4m?usp=sharing

Dataset used: Amazon Book review
Our dataset consists of 8.89 lakh reviews but due to computational constraints we have taken the first 50k reviews for our analysis.
Link of dataset used: http://jmcauley.ucsd.edu/data/amazon/

Module - 1 (Statistics):

1. **Explain the text processing pipeline adopted by you.**
   Our pipeline includes following steps:
   a) **Removing HTML tags** : Reviews may contain html and css markup, which needs to be preprocessed.
   b) **Case Folding** : Converting all characters into lower.
   c) **Removing punctuations** : Removing the punctuations.
   d) **Removing Stop Words of English** : There are stop words which occur multiple times in any type of document but they usually don't contain any information. So they need to be removed.
   e) **Lemmatization** : It is a process of converting terms/tokens into its actual dictionary format. Stemming, which is a heuristic process of chopping the end of a token, takes less time and computations. Stemming could have been performed but later we need to perform POS tagging, where passing the stemmed tokens will not produce any results.

2. Generate term statistics: [You can use nltk, Stanford parsers, spacy etc.]

a) Vocabulary size with word frequencies

Vocabulary contains the tokens which define a document in a corpus and not a stopword. We are taking the preprocessed dataset, which contains lemmatized reviews and splitting the reviews into tokens.

From the generated token streams, we are taking the frequency of each token and storing it in a form of python dictionary.

```
book : 100315
read : 38479
story : 35692
one : 32913
character : 25683
like : 21241
time : 20346
would : 16680
life : 16382
first : 15755
```

b) N-grams

It can be defined as a set of co-occurring tokens within a given window of size=n. Set of n-grams can be formed by taking n consecutive tokens together from a given document.

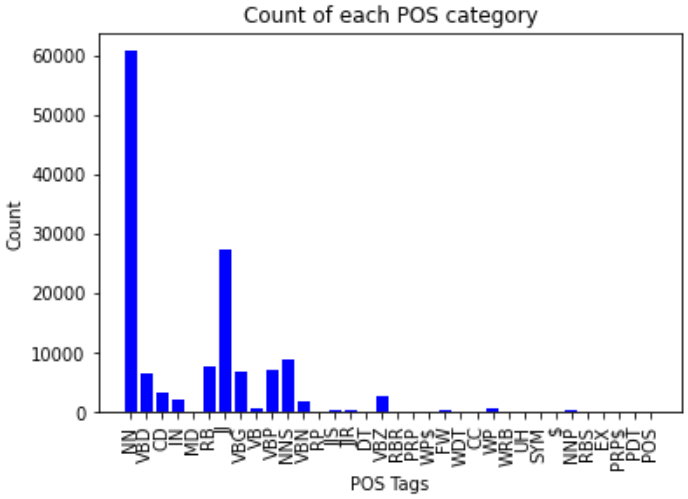
For example, if the sentence is **'I am a boy'** and **n=2**, then the set will be = **{('I', 'am'), ('am', 'a'), ('a', 'boy')}**

```
[['spiritually', 'mentally', 'inspiring'],
 ['mentally', 'inspiring', 'book'],
 ['inspiring', 'book', 'allows'],
 ['book', 'allows', 'question'],
 ['allows', 'question', 'moral'],
 ['question', 'moral', 'help'],
 ['moral', 'help', 'discover'],
 ['help', 'discover', 'really']]
```

c) POS Collections (Like Nouns - frequency, Verbs - frequency, Adverbs - frequency etc.)

POS stands for Parts-of-Speech. In English, there are various POS such as Noun, Pronoun, Adjective, Verb etc. From the corpus, we have extracted the POS using nltk's POS tagger.

frequencies of each type of POS :



d) <u>Most Frequent Noun Phrases</u>

In our corpus the most frequent Noun : book

e) <u>Most Frequent Verb Phrases</u>

In our corpus the most frequent verb : read

f) <u>NERs with their frequencies and types</u>

NER stands for Named Entity Recognition. Among the tokens, named entities will be those which are Proper Nouns, i.e. denoting a particular person, place, country, product etc. We have performed NER on the corpus using *Spacy*.

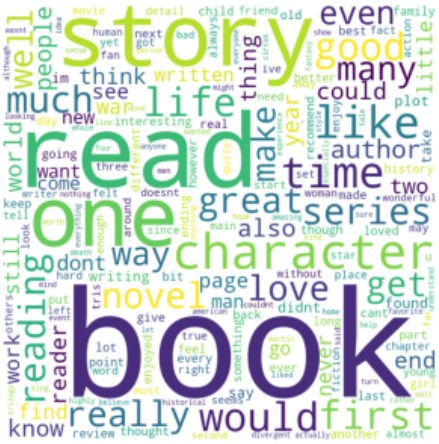
<u>3. Which set of terms best describe your corpus? How did you arrive at it?</u>

Top 10 most frequent terms from the corpus. These are the non-stopword terms found in the corpus most number of times and hence, these best describe the corpus.

```
book : 100315
read : 38479
story : 35692
one : 32913
character : 25683
like : 21241
time : 20346
would : 16680
life : 16382
first : 15755
```

4. Create a visualization and justify your answer for 3.

We have created a visualization using Word cloud. The most frequent words are the words having the largest size in the word cloud like Book, story, read, character, author, time, good, etc.



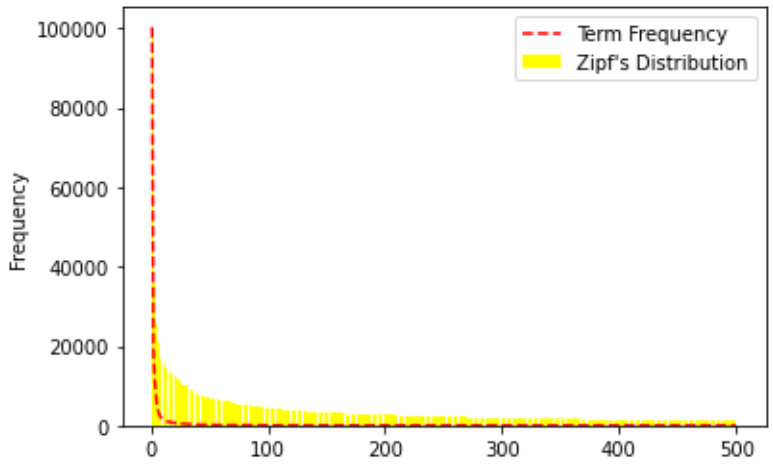
5. Plot a graph of the frequency of word vs rank of the word. How would you characterize the relationship? For your reference - consult "Zipf's law" – and determine the best fit for your corpus?

Zipf's law states that the frequency of a given term is inversely proportional with it's rank raised to the power of α where α≈1; $f \propto 1/r^{\alpha}$, where **f** is the frequency of the token, **r** is the rank of the token.

In Que 3, if we observe the 10 most frequent terms in our corpus, word **book** placed at rank 1 is having a frequency of 100315, read at rank 2 is having a frequency of 38479 and story at rank 3 is having a frequency of 35692.
The number of occurrences of the word **read** is roughly 0.38 times of the frequency of occurrences of the word **book** (which is close to 1/2).
The number of occurrences of the word **read** is roughly 0.35 times of the frequency of occurrences of the word **book** (which is close to 1/3).

The above observation validates Zipf's law.

<u>Frequency vs Rank plot (Zipf's law)</u>



Module - 2 (Topic analysis):

1. **Extract the topics from the corpus using LDA and present a visualization of them in your report. [You can use any ML library e.g Mallet]**

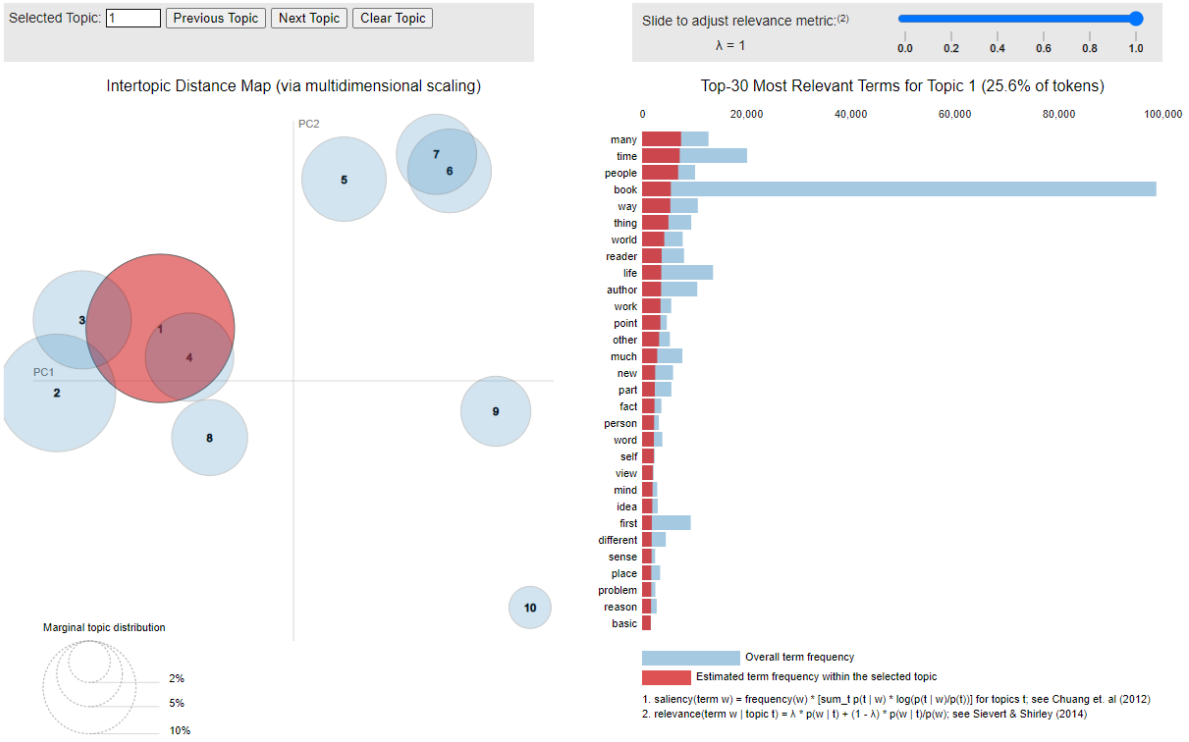
   Initially we have taken the number of topics to be trained using the LDA function of the gensim library to be 10.

   1.1. The 10 most important topics extracted from the reviewTextCorpus using WordCloud are:

In the word cloud of the Dominant Topic list,let's say for :

- **Topic 1** it consists of words like **recipe, cookbook** with highest probability of occurrence and we can conclude that it is related to cooking/cookery book reviews.
- **Topic 2** consists of words like **child,young,girl,family, woman, life** .Looking at these keywords, we can conclude that this topic is related to the family/characters aspect present in the story.
- Likewise **Topic 8** consists of important keywords like **rating, disappointed, star, bad, awful, intense, spoiler** which might be related to the reader's experience.

1.2. Visualization using pyLDAvis.gensim:



Here depending upon the inter cluster distance we can say that clusters 1, 2 3 and 4 are the most prominent and correlated clusters.

1.3. Perplexity and Coherence Score calculation

Now we have calculated the Perplexity and Coherence Score our lda_model.

Perplexity is a statistical measure of how well a probability model predicts a sample.Lower perplexity score (more negative value) implies a good topic model.

Coherence score measures how interpretable the topics are.These measurements help distinguish between topics that are semantically interpretable topics and topics that are artifacts of statistical inference.

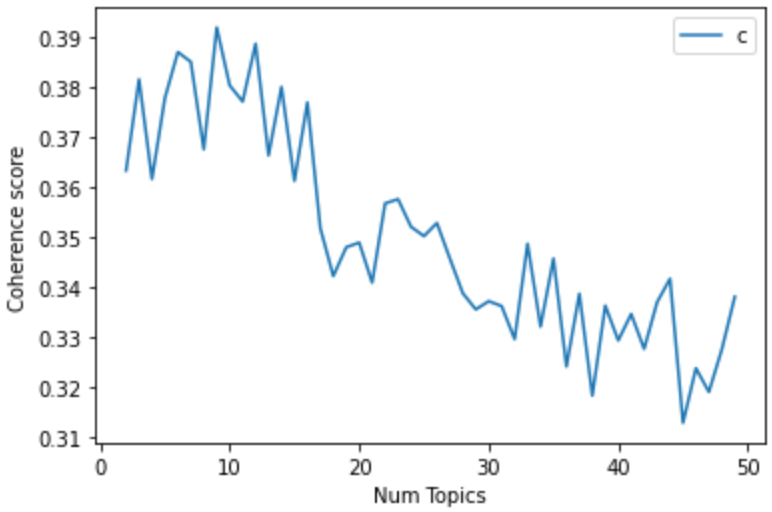
Coherence score is generally in between 0 to 1. For a good model the coherence score should be close to 1.

```
Perplexity:  -7.716029833460886

Coherence Score:  0.399160664338294
```

1.4. Number of Topics (from 2 to 50) Vs their Coherence Score

We have run one experiment where we have kept the number of topics from 2 to 50 and calculated their respective coherence score.



Output:

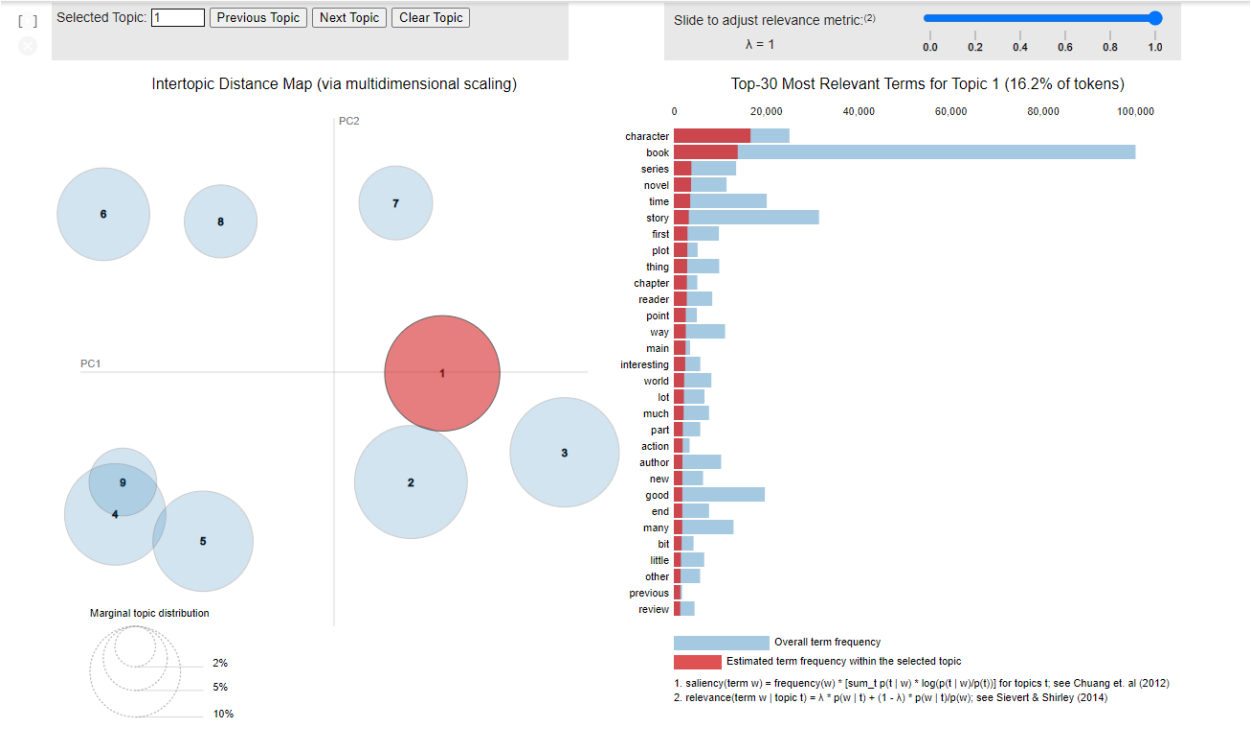
Num Topics = 2 has Coherence Value of 0.3633
Num Topics = 3 has Coherence Value of 0.3816
Num Topics = 4 has Coherence Value of 0.3616
Num Topics = 5 has Coherence Value of 0.3779
Num Topics = 6 has Coherence Value of 0.3871
Num Topics = 7 has Coherence Value of 0.3851
Num Topics = 8 has Coherence Value of 0.3676
Num Topics = 9 has Coherence Value of 0.392
Num Topics = 10 has Coherence Value of 0.3804
Num Topics = 11 has Coherence Value of 0.3772
Num Topics = 12 has Coherence Value of 0.3887
Num Topics = 13 has Coherence Value of 0.3664
Num Topics = 14 has Coherence Value of 0.3801
Num Topics = 15 has Coherence Value of 0.3613

Here we can observe that when the number of topics is 9, the coherence score is maximum i.e. 0.392, so we will be taking a maximum of 9 topics for our further analysis.

Most Important topics and relevant words with their probability:

```
[(0,
  '0.036*"recipe" + 0.015*"man" + 0.014*"old" + 0.012*"child" + 0.011*"woman" + 0.010*"young" + 0.010*"family" + 0.010*"year" + 0.009*"girl" + 0.008*"death"'),
 (1,
  '0.156*"book" + 0.035*"series" + 0.031*"story" + 0.029*"good" + 0.017*"character" + 0.017*"great" + 0.015*"first" + 0.014*"end" + 0.013*"time" + 0.011*"page"'),
 (2,
  '0.055*"character" + 0.046*"book" + 0.013*"series" + 0.012*"novel" + 0.012*"time" + 0.011*"story" + 0.010*"first" + 0.010*"plot" + 0.010*"thing" + 0.009*"chapter"'),
 (3,
  '0.017*"novel" + 0.010*"book" + 0.008*"many" + 0.008*"bread" + 0.008*"work" + 0.008*"word" + 0.007*"ingredient" + 0.006*"good" + 0.005*"food" + 0.005*"much"'),
 (4,
  '0.018*"book" + 0.009*"world" + 0.009*"people" + 0.009*"many" + 0.007*"time" + 0.007*"philosophy" + 0.007*"society" + 0.006*"history" + 0.005*"work" + 0.005*"year"'),
 (5,
  '0.038*"story" + 0.032*"life" + 0.016*"man" + 0.013*"novel" + 0.009*"reader" + 0.009*"tale" + 0.009*"time" + 0.008*"book" + 0.008*"human" + 0.008*"love"'),
 (6,
  '0.105*"book" + 0.028*"story" + 0.024*"time" + 0.019*"good" + 0.016*"great" + 0.013*"life" + 0.012*"people" + 0.011*"read" + 0.010*"thing" + 0.010*"many"'),
 (7,
  '0.039*"war" + 0.011*"great" + 0.009*"book" + 0.007*"novel" + 0.007*"man" + 0.007*"story" + 0.007*"people" + 0.007*"history" + 0.007*"life" + 0.006*"sea"'),
 (8,
  '0.048*"book" + 0.016*"good" + 0.010*"edition" + 0.009*"version" + 0.009*"word" + 0.008*"cooking" + 0.008*"time" + 0.007*"new" + 0.007*"page" + 0.007*"cookbook"')]
```

1.5. Visualization of the optimal Model



Here we observed that the prominent clusters have changed and now, 1, 2 and 3 are the most prominent clusters with most important keywords like book, character, novel, story. Other clusters are not so prominent because the count of words is very relatively less.

2. Present 20 most significant sentences for each topics

We have created a dataframe **df_dominant_topic** which consists of all the dominant topics along with their document number, topic percentage contribution, keywords and the representative texts. We have displayed the most significant sentence for each topic based on its topic percentage contribution.

Top 10 entries of the dataframe

	Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
0	0	7.0	0.4607	book, series, good, story, first, time, charac...	[inspiring, book, question, moral]
1	1	5.0	0.8875	many, time, people, book, way, thing, world, r...	[one, spirituality, literary, quality, message...
2	2	5.0	0.6692	many, time, people, book, way, thing, world, r...	[book, reflection, life, way, right, thing, sh...
3	3	5.0	0.3689	many, time, people, book, way, thing, world, r...	[book, revival, metaphysical, turbulent, profo...
4	4	1.0	0.4332	book, recipe, good, great, cookbook, cook, yea...	[timeless, classic, title, excellent, style, c...
5	5	3.0	0.5117	boy, bread, ingredient, day, game, white, old,...	[reading, mind, pool, water, cool, quiet, moss...
6	6	5.0	0.9357	many, time, people, book, way, thing, world, r...	[poetry, spiritual, visual, beauty, life, famo...
7	7	5.0	0.9357	many, time, people, book, way, thing, world, r...	[deep, dramatic, verse, heart, soul, truth, an...
8	8	4.0	0.4416	book, story, great, life, time, good, read, au...	[timeless, classic, year, gift, time, address,...
9	9	5.0	0.5388	many, time, people, book, way, thing, world, r...	[amazing, work, extensive, use, biblical, imag...

Based on the topic percentage contribution of representative texts for each dominant topic we have taken the list of the first 20 most significant text lists and stored the output in the excel file.

Link of the excel file:

https://docs.google.com/spreadsheets/d/14tZpvHk1Xsyiss9150LBR8EfoFE_J0gHV6rF5rBlcQE/edit?usp=sharing

The file consists of 9 topics each having top 20 sentences depending upon the topic percentage contribution.

Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
44202	1	0.9356999993	book, recipe, good, great, cookbook, cook, year, classic, easy,	['health', 'CATALOGUED', 'MORE', 'informative', 'book', 'health', 'benefit', 'food
43882	1	0.9308000207	book, recipe, good, great, cookbook, cook, year, classic, easy,	['today', 'recipe', 'book', 'lot', 'information', 'topic', 'lot', 'recipe', 'veggie', 'grant
49745	1	0.9308000207	book, recipe, good, great, cookbook, cook, year, classic, easy,	['book', 'great', 'product', 'lot', 'information', 'learning', 'tool', 'great', 'variety', '
48895	1	0.918200016	book, recipe, good, great, cookbook, cook, year, classic, easy,	['fabulous', 'collection', 'easy', 'recipe', 'food', 'group', 'collection', 'cook', 'sever
13647	1	0.918200016	book, recipe, good, great, cookbook, cook, year, classic, easy,	['year', 'single', 'volume', 'nice', 'great', 'price', 'nice', 'version', 'great', 'book']
49056	1	0.9100000262	book, recipe, good, great, cookbook, cook, year, classic, easy,	['worth', 'price', 'large', 'size', 'small', 'picture', 'info', 'well', 'book']
13799	1	0.9100000262	book, recipe, good, great, cookbook, cook, year, classic, easy,	['nice', 'quality', 'wonderful', 'literary', 'classic', 'book', 'collection', 'price', 'reas
19344	1	0.9100000262	book, recipe, good, great, cookbook, cook, year, classic, easy,	['vast', 'information', 'easy', 'fallow', 'instruction', 'sure', 'new', 'advanced', 'wat
26452	1	0.9100000262	book, recipe, good, great, cookbook, cook, year, classic, easy,	['student', 'date', 'great', 'condition', 'content', 'book', 'class', 'book', 'class']
14260	1	0.8999999762	book, recipe, good, great, cookbook, cook, year, classic, easy,	['good', 'condition', 'good', 'pricing', 'nice', 'cover', 'nice', 'yea']
48736	1	0.8999999762	book, recipe, good, great, cookbook, cook, year, classic, easy,	['gift', 'recipe', 'date', 'book', 'nice', 'picture', 'good', 'index']
49454	1	0.8999999762	book, recipe, good, great, cookbook, cook, year, classic, easy,	['book', 'helpful', 'word', 'many', 'word', 'letter', 'word', 'edition']
49154	1	0.8999999762	book, recipe, good, great, cookbook, cook, year, classic, easy,	['book', 'informative', 'purchase', 'money', 'price', 'free', 'shipping', 'thank']
48845	1	0.8999999762	book, recipe, good, great, cookbook, cook, year, classic, easy,	['favorite', 'restaurant', 'book', 'easy', 'follow', 'lot', 'fun', 'recipe']
3218	1	0.8999999762	book, recipe, good, great, cookbook, cook, year, classic, easy,	['nice', 'paperback', 'copy', 'collection', 'good', 'condition', 'good', 'price']
142	1	0.8999999762	book, recipe, good, great, cookbook, cook, year, classic, easy,	['book', 'decade', 'classic', 'word', 'book', 'copy', 'e', '-']
31588	1	0.8988000154	book, recipe, good, great, cookbook, cook, year, classic, easy,	['der', 'hilflo', 'vor', 'seinen', 'abgeschlachtet', 'ihm', 'dass', 'sich', 'der', 'junge', '
244	1	0.8874999881	book, recipe, good, great, cookbook, cook, year, classic, easy,	['purchase', 'book', 'repeat', 'good', 'informational', 'timely', 'resource']
47228	1	0.8874999881	book, recipe, good, great, cookbook, cook, year, classic, easy,	['book', 'precise', 'plenty', 'picture', 'good', 'condition', 'seller']

3. **Analyse whether the topics extracted make sense. Do you feel the topics cover the entire dataset correctly? Justify your claim. If you find anything is missing, please state what it is and why you think LDA might have missed them. Do you have any suggestions for improvement?**

LDA is a generative model, which assumes that the coherent documents are built around topic and and topics are built around words ,i.e closeness between words are captured by topics as a result of which the document itself can be regenerated. Using the LDA model our target is to recreate a document with some kind of coherence.

In our case we have taken the Amazon's books review dataset for training our model and due to computation limitations were able to train our model using only the first 50k review data even though the dataset consists of 8.85 lakh words. Since our review_text corpus is not very large, we noticed that the few of the words in few of the Topics didn't make complete sense and could have been wrongly classified, while a large group of keywords within a topic might be rightly grouped together.

If we observe the topics, we can conclude that these Topics did cover some of the aspects of books like category, reviews, genre, types of stories/plot ,attributes of characters, etc. but not all as in the LDA model we need to specify the number of topics ahead (static). For example if we take the keywords of Topic1:
Keywords: book, recipe, good, great, cookbook, cook, year, classic, easy, edition

And the list of most significant sentences (sentence in the form of list):
['health', 'CATALOGUED', 'MORE', 'informative', 'book', 'health', 'benefit', 'foods', 'herbs', 'recipes', 'health', 'benefit', 'treatment']
['today', 'recipe', 'book', 'lot', 'information', 'topic', 'lot', 'recipe', 'veggie', 'granted', 'recipe', 'simple']
['book', 'great', 'product', 'lot', 'information', 'learning', 'tool', 'great', 'variety', 'recipe', 'easy', 'follow']
['fabulous', 'collection', 'easy', 'recipe', 'food', 'group', 'collection', 'cook', 'several', 'year']
['year', 'single', 'volume', 'nice', 'great', 'price', 'nice', 'version', 'great', 'book']
['worth', 'price', 'large', 'size', 'small', 'picture', 'info', 'well', 'book']
['nice', 'quality', 'wonderful', 'literary', 'classic', 'book', 'collection', 'price', 'reasonable']
['vast', 'information', 'easy', 'fallow', 'instruction', 'sure', 'new', 'advanced', 'watercolorist']
['student', 'date', 'great', 'condition', 'content', 'book', 'class', 'book', 'class']
['good', 'condition', 'good', 'pricing', 'nice', 'cover', 'nice', 'year']
['gift', 'recipe', 'date', 'book', 'nice', 'picture', 'good', 'index']
['book', 'helpful', 'word', 'many', 'word', 'letter', 'word', 'edition']
['book', 'informative', 'purchase', 'money', 'price', 'free', 'shipping', 'thank']
['book', 'full', 'beautiful', 'picture', 'wide', 'variety', 'style']
['book', 'precise', 'plenty', 'picture', 'good', 'condition', 'seller']
['purchase', 'book', 'repeat', 'good', 'informational', 'timely', 'resource']
['nice', 'paperback', 'copy', 'collection', 'good', 'condition', 'good', 'price']
['favorite', 'restaurant', 'book', 'easy', 'follow', 'lot', 'fun', 'recipe']

Here we can see that the context of the topic is largely about the user's experience and review about the book, the information shared within the book, its price, seller, year, quality, etc.
Also this might be due to the small dataset used as well(document matrix used only 50k). If we would have been able to increase our training data size, the classification might have been more accurate and would have covered all the aspects related to Books as the books consist of a wide range of genres/stories.
Also due to resource limitations, we have trained our model for 100 iterations only, but LDA usually takes a long time to converge and we can infer that the keyword classification within a topic might be somewhat coherent but not optimal. To improve our result we can increase the training iterations of our model.