Emory University | MSBA, MACHINE LEARNING II

Topic Modeling of Amazon Reviews

Team “I like big data and I cannot lie” Derek Huang, Aman jiddewar, Mira Stefanova, Katie Wallis

Contents

[Case Overview 2](#_Toc5787870)

[Data Preprocessing 3](#_Toc5787871)

[Modelling 3](#_Toc5787872)

[Latent Semantic Analysis (LSA) 3](#_Toc5787873)

[Probabilistic Latent Semantic Analysis (PLSA) 4](#_Toc5787874)

[Latent Dirichlet Allocation (LDA) 4](#_Toc5787875)

[Topic Interpretation & Sentiment Analysis 5](#_Toc5787876)

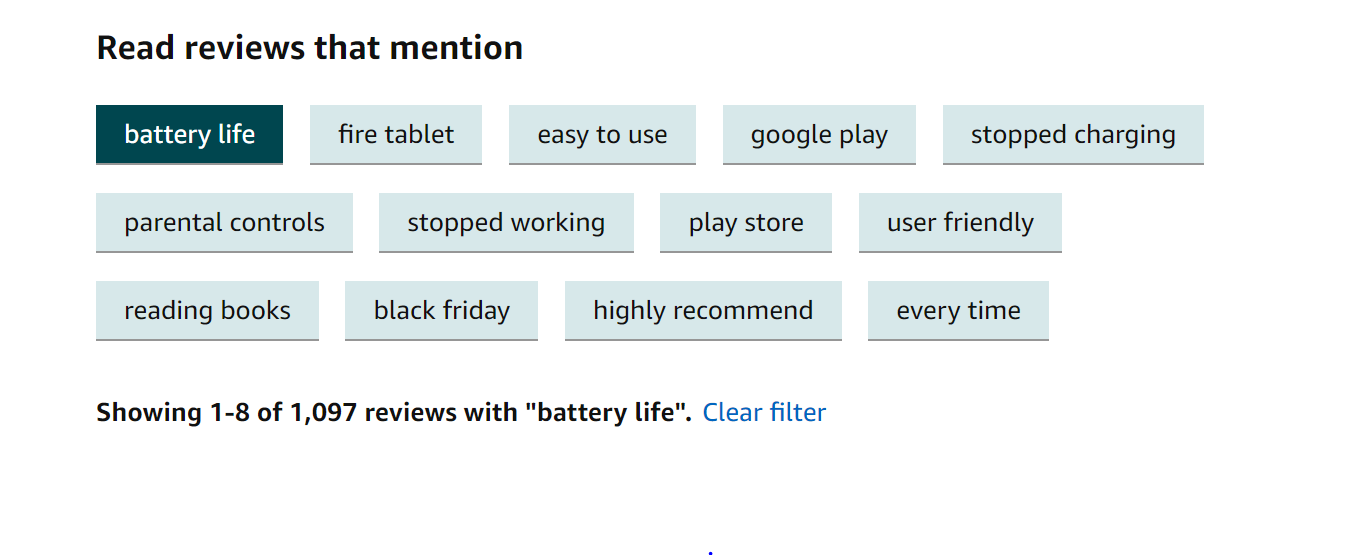
[Deployment 7](#_Toc5787877)

[Appendix 1. Extracting the positive and negative sentiment of each topic for Kindle 9](#_Toc5787878)

# Case Overview

In our last project for Managing Big Data, we used big data technologies to conduct sentiment analysis and discover n-grams for thousands of Amazon reviews. In this project, we would like to build upon our analysis and identify the most important topics mentioned in reviews of a given product.

For Amazon’s most popular products, there can be thousands of reviews. For customers who are interested in a particular aspect of a product, this can be overwhelming. They may not be able to find the reviews that cover what they are interested in. For sellers, having thousands of reviews decreases their ability to extract the feedback being given for improving their products. For this reason, there needs to be a way to filter reviews to more easily discover what reviews are relevant to you. Currently, Amazon’s solution to this problem comes from N-Grams analysis. As shown in the screenshot below, a user is able to filter reviews by clicking on the N-Gram that is more related to what they are looking for.



While this process is much better than nothing, it is imperfect. Some of the N-Grams don’t make much sense to filter by. For instance, “fire tablet” doesn’t narrow down many reviews because the reviews are for the Kindle Fire, which is a tablet. Similarly, filtering by “every time” doesn’t make much sense either as the customer isn’t perfectly sure what the reviews would be referring to. The reviews themselves that fit under that filter may not even be very homogenous.

Instead, we think we can improve the experience for both the customer and the seller by extracting topics from the reviews. Our envisioned functionality would be the same as the N-Grams, but the choices to select from would make more sense.

For the scope of this project, we examined reviews from a few products within the Kindle Fire family and conducted topic modelling using LDA, LSA and PSLA models. Once the models provided us with groups of words, we manually assigned topics to them. We wanted to go one step further to identify which topics were around product issues and which were around product strengths. To do this, we found the average sentiment of reviews that contained words related to the relative topics.

Ultimately, our analysis found discovered 5 key topics for the Kindle Fire from the various modelling techniques: Screen Readability and Responsiveness, Purchasing Logistics, Gift, Camera and Battery.

# Data Preprocessing

To make our upcoming modelling more efficient and effective we first cleaned up the reviews data by preprocessing it through a combination of commonly accepted steps of preprocessing for natural language processing tasks and some additional steps fitting to the specific data set and business use case. All the preprocessing steps of the reviews text include:

* tokenizing the reviews;
* normalizing all words to lowercase;
* removing stop words;
* removing words shorter than 3 letters or longer than 10 letters, as those will most probably be typos;
* stemming the words;
* removing words that are too common, such as the brand and product names, as those would not be valuable distinguisher across review;

Furthermore, we looked at the role of different words in the sentences and identify nouns, adjectives and verbs. We iterated over the modelling approached explained in the next section first using only the nouns of reviews as input data; then using nouns and adjectives as input data and lastly using nouns, adjectives and verbs. The resulting underlying topics and corresponding words made the most sense when using all three types of words and therefore the final modelling results presented in the following sections are based on input data which includes nouns, adjectives and verbs.

# Modelling

After data preprocessing, we explored three models to better capture the latent topics: Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), and Latent Dirichlet Allocation (LDA). All of those three models were trained on the Kindle Family to extract the topics so that we can better understand the relationship between topics and reviews.

## Latent Semantic Analysis (LSA)

LSA is a basic model employs Singular Value Decomposition with document-words as input and the topics with respect to each word and documents as output. By choosing the number of topics, we decide the elements in the V and U matrix, which can be interpreted as the relationship between topics and documents or the words.

The LSA has one very strong advantage is that it is extremely efficient when applying. The reason is that it uses singular value decomposition so that the return is based simply on the decomposed matrixes. When the immediate topic extraction is required, it is one good method. However, it has multiple cons that need to be taken care of. The first is that it lacks the interpretable embedding, where the coefficients associated with each word have two signs. Statistically speaking, negative sign refers to a lack of the word. However, when analyzing it within real-world context, it tends to be not interpretable. Meanwhile, it requires a relatively large data set to be accurate.

The number of topics is selected based on the coherence score shown below. The number associated with the optimal occurrence score is three, which is the input for a number of topics.

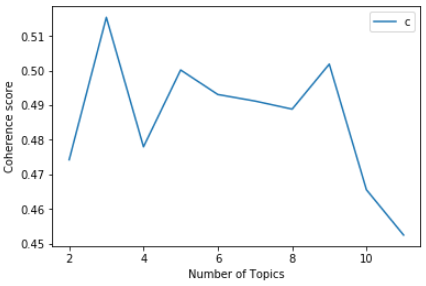


Figure LSA coherence score across the different number of topics. (Note: The higher the better)

## Probabilistic Latent Semantic Analysis (PLSA)

PLSA is a probability version of LSA, which takes consideration of the conditional probability of document given topic and probability of words given a topic. Instead of returning coefficients, it returns the conditional probability for the U and V matrix.

The PLSA is more flexible than LSA given that it involves the word and document conditional probability. However, it is even harder to explain the results from PLSA because the numbers are the conditional probabilities which lack the interpretation in the matrix. Moreover, as a probability-based model, it tends to overfit with the increasing number of documents in the input data. Another disadvantage is that when new data come in since there is no parameter for the probability function.

For PLSA, it is hard to determine the number of topics since coherence sore does not work well on this model. Our team manually experimented with different numbers and set the number as five since four topics among them is interpretable based on the words with top conditional probability.

## Latent Dirichlet Allocation (LDA)

The model that is most popular currently is LDA modelling. It is a Bayesian version of PLSA using Dirichlet priors for the document-topic and word-topic distributions. According to Wikipedia, it allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics (https://en.wikipedia.org/wiki/Latent\_Dirichlet\_allocation).

It has numerous advantages comparing to the previous two models. The first is that it can be better generalized than the PLSA model, which means that when new data points are input to the model, it can generate the relating vector representing their topic mixtures in coefficients. It is more accurate than both PLSA and LSA, and easier to interpret since each topic is characterized by the words they are most strongly associated with. This is due to the fact that all the coefficients responding to each word are all positive numbers. As a result, the larger the value is, the more strongly the word is relating to the topic. However, it is slower in training process and still requires human intervention to label the topic.

The number of topics is also determined by the coherence score, where the optimal score is achieved when the number of topics equals three.

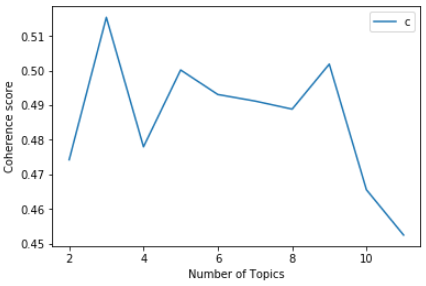


Figure LDA coherence score across the different number of topics. (Note: The higher the better)

# Topic Interpretation & Sentiment Analysis

Once the models had given us the best topic groups of words, we manually interpreted them. This did not come as an easy process. Some of the topics did not make a lot of sense and many of the same words such as “screen” seemed to be present in several topics. After a few iterations of the model building process, we discovered groupings of words from several models that made sense.

Here are the initial groupings discovered by each model.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | LDA | LSA | LSA | PSLA | PSLA | PSLA | PSLA |
| Topic | **Screen** | **Battery** | **Screen/Readability** | **Price** | **Gift** | **Screen** | **Camera** |
| Words | product | buy | read | Everything | read | love | Camera |
| buy | charge | app | hop | gift | print | look |
| page | app | book | offer | buy | think | take |
| screen | screen | page | money | use | recommend | picture |
| turn | issue | work | own | book | book | price |
| game | work | charge |  | Christmas |  |  |
| price | charger | issue |  |  |  |  |
| click |  | screen |  |  |  |  |
| app |  | reader |  |  |  |  |
| touch |  |  |  |  |  |  |

Table Overview of the most meaningful topics and corresponding words from each of the three models

From these groupings, we wanted to combine some of the categories and to use business understanding to narrow down the words that made up the different categories. After we combined the columns, we were left with 5 main topics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen Readability and Responsiveness** | **Purchase Logistics**  **(Price and Delivery)** | **Gift** | **Camera** | **Battery** |
| page | problem | gift | Camera | buy |
| screen | buy | buy | look | charge |
| click | order | game | take | screen |
| touch | charge | Christmas | picture | work |
| print | receive |  |  | charger |
| read | day |  |  |  |
| reader | offer |  |  |  |
|  | money |  |  |  |

Table Overview of defining words of the final 5 main topics across models

While we did take some creative liberty in the interpretations, we believe these topics do a good job of describing the main topics in the reviews. The resulting topics come from a census of the important topics discovered across several models.

Once we had clear interpretations of our topics, we wanted to know how well the product was performing along these dimensions. To do this, we attempted to map reviews to different topics and find the average sentiment for the topic overall. The mapping of reviews to the topic wasn’t easy to do perfectly. We decided that a review would be considered to cover a topic if one of the key identifying words from that topic were included in the review. We did not map back all of the words that the model provided because some of the words were less directly related to the topic. For instance, if we mapped back every review that uses the word “book” isn’t necessarily talking about the screen or readability of the tablet. However, if a review uses the word “screen”, “click” or “touch” they probably are referring to the “Screen Readability and Responsiveness” category.

In addition to the sentiment of topics, we wanted to compare how prevalent each topic was. To do this, we counted the number of reviews that were mapped to each topic. It is worth mentioning that a given review may be linked to several topics. For instance, a review that says “I love the battery life of my kindle, but the screen is sometimes hard to read” would be mapped back to the screen and battery categories. We believe this makes sense from a business understanding point of view.

When we roughly plotted the frequency of mentions with the positive sentiment, we determined that areas like “Purchasing Logistics” and “Battery” were the most pressing problems because they had the lowest sentiments and the highest frequency of mentions. While “Gifts” and “Screen” had high sentiments, but the only average frequency. Lastly, the camera category is noteworthy because it has a decent sentiment, but low frequency. We believe this is because different Kindle versions are being combined here. The older kindles didn’t have cameras, so cameras wouldn’t be mentioned as often overall. While we did not ideally want to combine different versions of categories, there weren’t enough reviews of a single product alone.

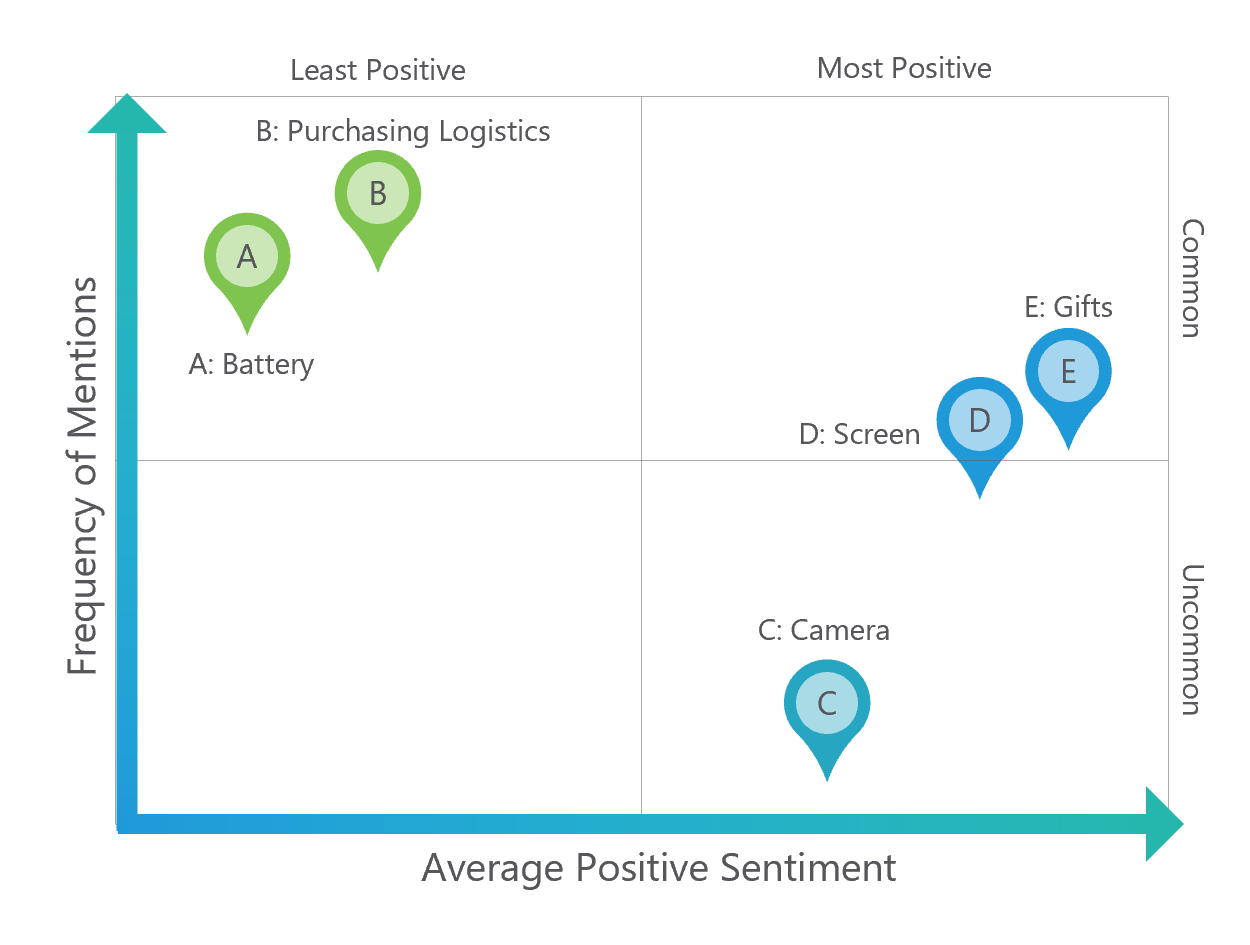


Figure Positioning of each of the 5 topics across the grid of average positive sentiment and frequency of mentions

# Deployment

Our project has been a proof of concept and the next step would be to generalize and deploy this approach to all products. The main deployment challenge we have identified for achieving our desired business use cases with this modelling approach is that topic labels will need to be manually assigned for each of the millions of products that Amazon has. The following steps are a potential solution we propose for automatically generating topic labels for product features:

1. Input data of each category of product (for example, reviews for all tablets) and used LSA and LDA to generate topics
2. Generating labels by associating each topic with 2-5 n-gram words (since n-gram captures nouns better, which can be used as attributes of the products), assign weights based on all the reviews under the n-gram words.
3. Rank each review based on the coefficients of terms for each topic-label
4. For both the topic label and review, create the helpful click so that customers can rate if the topic or the review is helpful.
5. Use the number of helpful clicks for a topic to update weights for topic-labels and keep only the dominate labels.
6. Update the ranking score based on the number of helpful clicks for each review and re-rank reviews under each topic based on the score

Furthermore, the accuracy and effectiveness of topical modelling depend on the amount of reviews data available. We were not able to find reliable research suggesting a specific threshold of the amount of data necessary for effective topical modelling. We, therefore, suggest that a further investigation of the validity of the results produced by performing this modelling on the lower quartile of products in terms of their number of reviews.

# Appendix 1. Extracting the positive and negative sentiment of each topic for Kindle

Our team has already performed sentiment analysis of each review in this dataset during the Managing Big Data project, enriching the data on each review with three columns - percentage positive sentiment, percentage neutral sentiment and percentage negative sentiment. Below we present the queries used to extract the sentiment for each of the five final topics for the Kindle family.

CREATE EXTERNAL TABLE IF NOT EXISTS reviews\_df.reviews\_dataframe (

`id` int,

`product\_id` string,

`star\_rating` string,

`total\_votes` int,

`helpful\_votes` int,

`review\_body` string,

`possentiment` double,

`neusentiment` double,

`negsentiment` double,

`reviews` string

)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'

WITH SERDEPROPERTIES (

'serialization.format' = ',',

'field.delim' = ','

) LOCATION 's3://reviewsdf/'

TBLPROPERTIES ('has\_encrypted\_data'='false');

**# Screen Readability and Responsiveness**

select product\_id, count(review\_body), avg(possentiment) from reviews\_dataframe

where reviews like '%page%' or reviews like '%click%' or reviews like '%touch%' or reviews like '%screen%' or reviews like '%print%' or reviews like '%read%'or reviews like '%reader%'

group by product\_id;

**# Gifts**

select product\_id, count(review\_body), avg(possentiment) from reviews\_dataframe

where reviews like '%gift%' or reviews like '%buy%' or reviews like '%use%' or reviews like '%Christmas%' or reviews like '%play%'

group by product\_id;

**# Purchasing logistics**

select product\_id, count(review\_body), avg(possentiment) from reviews\_dataframe

where reviews like '%problem%' or reviews like '%product%' or reviews like '%buy%' or reviews like '%order%' or reviews like '%charge%' or reviews like '%recieve%'or reviews like '%day%'or reviews like '%offer%'or reviews like '%money%'or reviews like '%own%'

group by product\_id;

**# Battery**

select product\_id, count(review\_body), avg(possentiment) from reviews\_dataframe

where reviews like '%buy%' or reviews like '%charge%' or reviews like '%app%' or reviews like '%screen%' or reviews like '%issue%' or reviews like '%work%'or reviews like '%charger%'

group by product\_id;