Latent Dirichlet Allocation Models Considering Emojis

Taikgun Song

Abstract

Latent Dirichlet Allocation (LDA) is a popular topic modeling technique for natural language processing. Emoji is a pictogram that is commonly used on Social Networking Service (SNS) such as Twitter and Instagram. Conventionally, emoji characters were deleted before applying LDA to text data from SNS. In this research, we compared the performance of the LDA under three different cases: (1) LDA with text data including emojis as unicode characters, (2) LDA with emoji characters excluded from the data, and (3) LDA after trasnalting the emoji characters into English. Using Twitter text messages of three different hastags, our analyses revealed that case (1) LDA with text data including emojis as unicode characters identified the topics the best. Our result highlights that an emoji character aggregates information of n-gram words into a uni-gram word which significantly enhances the performance of LDA by preventing the loss of information during the exclusion or translation process.

Emoji characters have an increasing role in our interactions on social media, allowing us to express a range of feelings in just a single (Unicode) character. This makes emojis a rich source of information in textual analyses. We compare the performance of Latent Dirichlet Allocation (LDA) in the presence of emojis. We explore the impact of emojis by conducting LDA in three different scenarios: (1) Standard practice in LDA is to delete any non ASCII characters (including emojis), (2) LDA with Emojis left in their raw text form, and (3) LDA after translating the emoji characters into their English counterpart. Using Twitter text messages from three different hashtags, our analyses reveals that case (2) LDA with Emojis left in their raw text form identifies these topics the best. Our results highlight that emoji characters are similar in information to selected n-gram words and the loss of information by deleting emojis or translating them into (multi-gram) English words significantly reduces the LDA performance.

Contents

1 Introduction

Text data contains valuable insights that is useful for content recommendation, customer care service, social media analysis, and others. However, the information is usually hidden within the text and has to be extracted using a modeling approach. Topic modeling is a text-mining method that extracts information from a text by identifying latent semantic structures in the text body. One of the most widely used topic modeling methods is the Latent Dirichlet Allocation(LDA). LDA is a hierarchical Bayesian model which assumes that

each of the documents in a collection consists of a mixture of topics, and these topics are responsible for the choice of words in each document. Topics, are the latent part of the document set and one can only observe words collected in the documents. LDA uses statistical inference to discover structure given the words and documents by calculating the relative importance of topics in documents and words in topics.

The rapid growth in internet and telecommunication technology triggered the development of Social Network Services(SNS) platform such as Tweeter, Facebook, and blog posts. The SNS messages often include individual's perceptions, feelings, and opinions. Evaluating this data may be meaningful for policy makers, social science researchers, and business entrepreneurs. This electronic word-of-mouth heavily uses text data as the medium of communication. Thus, topic modeling including LDA may be ideal method for analyzing SNS text data for information retrieval tasks.

The use of emoji - a pictogram that expresses the author's feeling and emotion - mixed in with other text is a unique characteristic of SNS messages that distinguishes itself from other text data. As shown in Figure 1, many SNS messages can be found with emoji embedded in the content. Conventionally, emoji characters have been considered as a noise and were deleted prior to applying LDA techniques and other topic modeling methods. Nevertheless, one should focus on the richness of information that emoji characters can provide. Especially consider the emotional and symbolic representation of emoji that cannot be better expressed with alphabet characters. Therefore, in contrast to the typical topic modeling procedure, this paper proposes the idea of incorporating emoji characters to enhance the performance of the LDA method on SNS text data.

Figure 1: Example of Twitter Messages

The use of emoji characters has three main benefits. First, it reduces the systematic problem of LDA with data sparsity. All emoji characters have name and keywords associated with the contextual meaning that it conveys. By translating emoji characters into English text and related keywords increases the amount of the text observed, and thus leads to better LDA results. Second, each emoji character has a set of pre-determined topic dimension assigned to it by the official organization. This information can be used as auxiliary information during the topic matching process. Lastly, the emoji character itself is an abstract of emotion and symbolic representation. Thus, it is natural to take the output of LDA containing emoji

translation to sentiment analysis.

2 Latent Dirichlet Allocation (LDA)

At its core Latent Dirichlet Allocation is a generative statistical model, that identifies posterior probabilities of words belonging to previously unidentified topics, and topics belonging to documents. The underlying model is generally not analytically tractable, but uses a Gibbs sampling approach instead. Here, we are deriving the posterior distributions involved in more detail. A graphical overview of LDA is given in Figure 2.

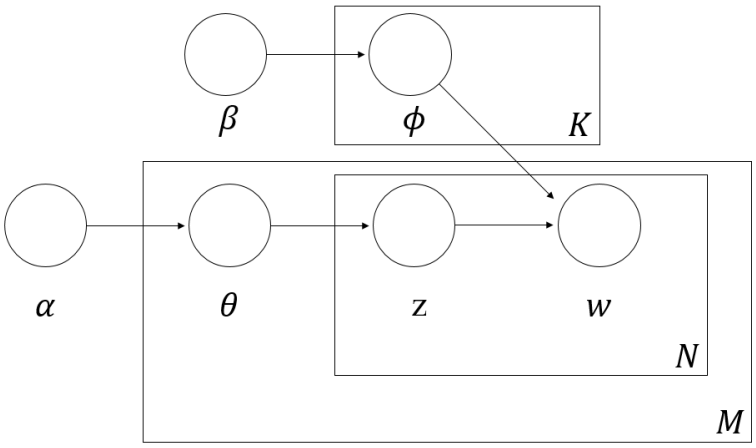


Figure 2: Graphical model representation of LDA in plate notation.

Let M be the total number of documents in the data set, and N_m be the number of words in the m^{th} document. Let K be the total number of topics in the data. Define w_{mn} be the n^{th} word in the m^{th} document. LDA assumes the distribution of w_{mn} to follow a Multinomial distribution with parameter $\phi_{z,w}$. $\phi_{z,w}$ is a probability of observing word w in topic z. The model assumes that the distribution of words in topic z, i.e., $\boldsymbol{\phi}_z$, follows a Dirichlet distribution with prior $\boldsymbol{\beta} = \begin{bmatrix}\beta_1 \cdots \beta_N\end{bmatrix}$. Let $z_m n$ be the topic of assigned to the word w_{mn}. Then, the model assumes $\boldsymbol{z_m}$ to follow a Multinomial distribution with parameter $\boldsymbol{\theta}_m$, where $\boldsymbol{\theta}_m$ is the distribution of topics in document m. The distribution of $\boldsymbol{\theta}_m$ is assumed to follow a Dirichlet distribution with a prior $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_1 \cdots \alpha_K\end{bmatrix}$.

Let w_{mn} be the n^{th} word in the m^{th} document. We assume that the topic of w_{mn} is z_m, a topic associated with document m. Assume $z_m \sim Multinomial(\boldsymbol{\theta}_m)$, where $\boldsymbol{\theta}_m \sim Dirichlet(\boldsymbol{\alpha})$ for all $m = 1, \dots M$ and $\alpha > 0$. For a given topic $z_m = k$, we assume that $w_{mn} \sim Multinomial(\boldsymbol{\phi}_k), n = 1, \dots, n_m, m = 1, \dots, M$, where $\boldsymbol{\phi}_k \sim Dirichlet(\boldsymbol{\beta})$, $k = 1, \dots, K$.

The summarization of the assumptions are written below.

1. M: The total number of documents in the data set
2. N_m: The number of words in the m^{th} document
3. K: The total number of topics in the data set
4. w_{mn}: n^{th} word in document m, $m \in \{1, \dots, M\}$ and $n \in \{1, \cdots, N_m\}$
5. z_{mn}: The topic of the w_{mn}, $z_{mn} \in \{1, \dots, K\}$
6. $\boldsymbol{\alpha}$: A vector of prior weights for each topic in a document
   $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_1 \cdots \alpha_K\end{bmatrix}$

7. $\theta_{m,k}$: The probability of observing topic k in document m
   $\boldsymbol{\theta}_m \sim Dir(\boldsymbol{\alpha})$: The distribution of topics in document m
   $$\boldsymbol{\theta}_{M\times K} = \begin{bmatrix} \boldsymbol{\theta}_1 = (\theta_{1,1}, \theta_{1,2}, \cdots, \theta_{1,K}) \\ \boldsymbol{\theta}_2 = (\theta_{2,1}, \theta_{2,2}, \cdots, \theta_{2,K}) \\ \vdots \\ \boldsymbol{\theta}_M \end{bmatrix}$$
8. $\boldsymbol{\beta}$: A vector of prior weights of the word distribution for each topic
   $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \cdots \beta_N \end{bmatrix}$
9. $\phi_{z,w}$: The probability of observing word w in topic z
   $\boldsymbol{\phi}_z \sim Dir(\boldsymbol{\beta})$: The distribution of words in topic z
   $$\boldsymbol{\phi}_{K\times N} = \begin{bmatrix} \boldsymbol{\phi}_1 = (\phi_{1,1}, \phi_{1,2}, \cdots, \phi_{1,N}) \\ \boldsymbol{\phi}_2 = (\phi_{2,1}, \phi_{2,2}, \cdots, \phi_{2,N}) \\ \vdots \\ \boldsymbol{\phi}_K \end{bmatrix}$$
10. $z_{mn} \sim Multinomial(\theta_m)$
11. $w_{mn} \sim Multinomial(\phi_{z_{mn}})$

Then, the total probability of the model is given as the product of the conditional probabilities

$$p(W, Z, \theta; \phi, \alpha, \beta) = \prod_{i=1}^{K} P(\phi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t}|\theta_j) P(W_{j,t}|\phi z_{j,t})$$

The marginal distribution of word w given hyper parameter α and β is then obtained by integrating the below equation:

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{v=1}^{V} \sum_{z_v} p(z_v|\theta) p(w_v|z_v, \beta) \right) d\theta$$

The posterior distribution is given as the following equation, however, it is intractable for exact inference and Gibbs sampling is used to infer the variables.

$$p(\boldsymbol{\theta}, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)}$$

3 Data preparation

3.1 Removing Stop Words

A natural language can be categorized as two distinctive set of words: content/lexical words and function/structure words. Content/lexical words are words with substantive meanings. Function/structure words on the other hand have little lexical meaning, but establish grammatical structure between other words within a sentence.

LDA models a document as a mixture of topics, and then each word is drawn from one of its topic. Therefore, the method depends on the frequency of observed words in a given text data set. This makes LDA vulnerable to high frequency function/structural words. Thus, any group of non-informative words including the function/structural words should be filtered out before doing an analysis. This group of words is called `stop words`. For example, prepositions(of, at, in, without, between), determiners(the, a, that, my), conjunctions(and, that, when), pronouns(he, they, anybody, it) are common examples of the `stop words`. For the analysis done here, the `tm` package in `R` was used to delete the stop words.

	Original Tweet	Tweet with Stopword Removed
1	loving this misty weather this sweater and my favorite couple	loving misty weather, sweater favorite couple
2	fairytale atmosphere in alberobello Let's go for a walk	fairytale atmosphere alberobello Let's go walk
3	Me when ashleytisdale puts a New music session on YouTube	Me ashleytisdale puts New music session YouTube

Table 1: Example of removing stop words using the Twitter data

3.2 Stemming

Due to structural and grammatical reasons of English, a family of words that are driven from a single root word is used in different forms. For example, words such as "stems", "stemmer", "stemming", and "stemmed" are all based on the root "stem". Words with the same meaning but different forms contribute to data sparsity, reducing the performance of the LDA method. `Stemming` cuts inflectional forms of a word to its root form and increases the frequency of observed stems.

Stemming has two disadvantages. First, there is the possibility of over stemming. For example, three different words "universal", "university", and "universe" have the same stemmed word "univers". The accuracy of the LDA method may decrease by putting words with different meanings into a single topic. Moreover, when the LDA output is given as a stemmed word, it is difficult to trace the stemmed word back to its original form. To overcome this problem, this paper matched the stemmed word to the most frequently used original word. Example of stemming using the `tm` is provided in Table 2.

	Original Tweet	Tweet after Stemming
1	loving this misty weather, this sweater and my favorite couple	love this misti weather, this sweater and my favorit coupl
2	fairytale atmosphere in alberobello Let's go for a walk	fairytal atmospher in alberobello Let go for a walk
3	Me when ashleytisdale puts a New music session on YouTube	Me when ashleytisdal put a New music session on YouTub

Table 2: Before and after Stemming

3.3 n-gram

n-gram is a neighboring sequence of n items from a collection of text data set. This item could be anything from phonemes or syllables to letters or words based on the application. Applying the concept of n-gram is important in computational linguistics is important especially with LDA, since n-gram is used as part of the prior distribution.

An example of word-level-n-gram with text "he is a nice person" is given in Table 3.

1-gram (unigram)	2-gram	3-gram	4-gram	5-gram
he	he is	he is a	he is a nice	he is a nice person
is	is a	is a nice	is a nice person	
a	a nice	a nice person		
nice	nice person			
person				

Table 3: Example of word-level-n-gram

Moreover, n-gram approach can help identify misspelled words or out-of-vocabulary words that commonly exist on the online platform. For example, the distance of the letter-level n-gram could be used to match strings.

4 Application

4.1 Data Set and exploratory data analysis

Three samples of Twitter messages with the following hash-tag #inlove, #hateher, and #marchscience were scraped. The data set contains 944 #inlove messages, 1145 #hateher messages, and 1195 #marchscience messages. The proportion of Twitter messages containing emoji characters per hashtag is illustrated in Table 4. 52.7% of the #inlove tweets, 29.3% of the #hateher tweets, and 7.8% of #marchscience tweets make use of one or more emojis.

Table 4: Proportion of Twitter messages with emoji

	#inlove	#hateher	#marchscience
Proportion	0.5275	0.2926	0.07782

For the hashtag #inlove, a total number of 1188 emojis were used, consisting of 182 unique emojis. For hashtag #hateher, 695 emojis from 112 unique emojis were used. For hashtag #sciencemarch, 202 emojis from 102 unique emojis were used. Top 5 frequently used emojis per hashtag is given in Table 5. It was interesting to see "Face with tear of joy" as the most popular emoji for hashtag #hateher. Although the name itself contains the word "joy", some users of this emoji adopted this pictogram to express their mixed feeling of love and hate at the same time.

#inlove	emoji	Count	#hateher	emoji	Count	#marchscience	emoji	Count
U+1F60D	😍	297	U+1F602	😂	154	U+1F52C	🔬	13
U+2764	❤️	164	U+1F644	🙄	88	U+1F30E	🌎	11
U+1F495	💕	47	U+1F621	😡	40	U+1F44D	👍	9
U+1F618	😘	40	U+1F612	😒	38	U+1F680	🚀	8
U+2728	✨	26	U+1F62D	😭	36	U+1F30D	🌍	7

Table 5: Five most popular emoji for each hashtag

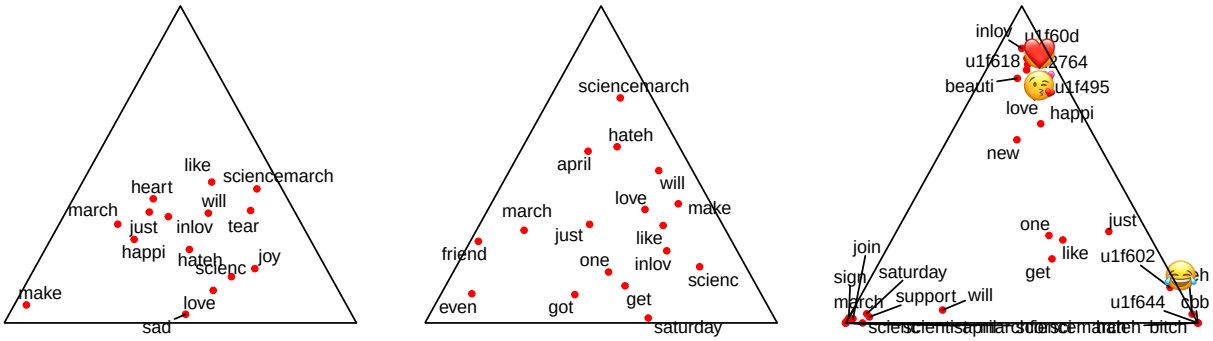
Two hashtag #inlove and #hateher were selected since the two topics should contain the opposite sentiments. The hastag #sciencemarch was used as a control topic, thus making all hashtags distinctive. If the performance of the LDA is effective, then the composition words assgined to each latent topic should have similar characteristics based on the three initial hastags.

4.2 Results

LDA was performed on the following three difference cases:

1. LDA on a data set with emoji characters deleted
2. LDA on a data set with emoji characters as unicode
3. LDA on a data set with emoji translated into English

A ternary plot is a triangular graph that displays three variables with respect to its proportion that sum to one. Ternary plots were used to illustrate the LDA ouput of the three different cases listed above. Each corner of the triangle represents the topic determined by the LDA method. The conditional probability of a topic given a word was calculated for all words in the corpus. The calcualted conditional probability was used as the proportion to construct the ternary plot.



(a) Emoji translated in English (b) All emoji characters excluded (c) All emoji characters included

Figure 3: LDA Output displayed as a ternary plot for each methods

The first Figure 3a have words clustered at the center of the ternary plot. The second Figure 3b have words spreaded our more than Figure 3a, but not as much as the Figure 3c. The last Figure 3c have four distinctive clusters. Three at each end point of the triangle, and a small cluster at the center. The ternary plot in 3 indicate that the performance of the LDA on the last 3c was the best since it successfully assigned words with similar characteristics to each laten topics, and a group of general words at the center of the ternary plot.

4.2.1 LDA with emoji characters deleted

The current standard practice in natural language processing removes emoji characters from the text data. For the first case, therefore, all emoji characters provided as unicode were removed from the data set before performing LDA. The result of the LDA with three topic dimensions is provided in Table 6. The bar-chart in Figure 4 demonstrated that the LDA method did not successfully distinguish the three topics well. The words in Topic 1 turned out to be a mixture of 'Hate Her' and 'Science March', Topic 2 had all three topics, and Topic 3 was associated with 'Hate Her' and 'In Love'. Ternary plot constructed in 3b showed that the output from the LDA was located at the center of the ternary plot. This indicates that words are weakly connected to all topic dimensions supporting that LDA was not effective.

Table 6: LDA output of conditional probability of word given topic when emoji characters are deleted

1.term	1.phi	2.term	2.phi	3.term	3.phi
hateh	0.0379	sciencemarch	0.0393	hateh	0.0523
sciencemarch	0.0300	inlov	0.0241	inlov	0.0506
scienc	0.0131	love	0.0181	sciencemarch	0.0178
inlov	0.0093	scienc	0.0152	love	0.0104
march	0.0092	just	0.0127	scienc	0.0069

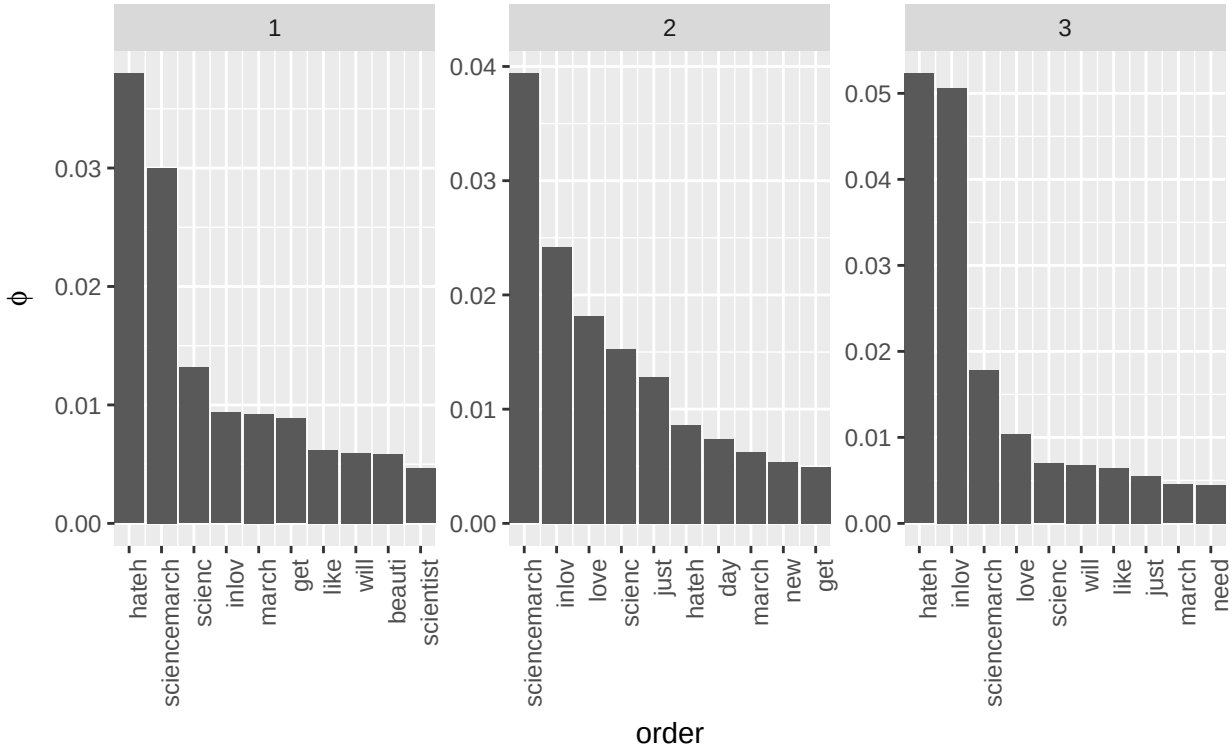


Figure 4: Graphical display of Table 6

4.2.2 LDA with emoji translated in English

Each emoji characters can be translated into a multi-gram English word. Thus, LDA was practiced on the data with English translated version of the emoji characters. The output of LDA with translation is provided in Table 7. The bar-chart in Figure 5 revealed that the performance of the LDA method was ineffective. Words from 'Hate Her' and 'Science March' were assigned to Topic 1, top words in Topic 2 were from 'In Love', 'Hate Her', and 'Science March', and top words in Topic 3 were composed with words from 'Hate Her' and 'In Love'. The ternary plot in 3a also showed that there were no strong relationship between the words and a specific topic.

Table 7: Conditional probability of word given topic when emoji characters are translated into English

1.term	1.phi	2.term	2.phi	3.term	3.phi
hateh	0.0428	sciencemarch	0.0256	hateh	0.0425
heart	0.0407	scienc	0.0178	sciencemarch	0.0391
inlov	0.0323	heart	0.0130	inlov	0.0344
sciencemarch	0.0154	inlov	0.0108	march	0.0111
scienc	0.0123	march	0.0090	just	0.0096

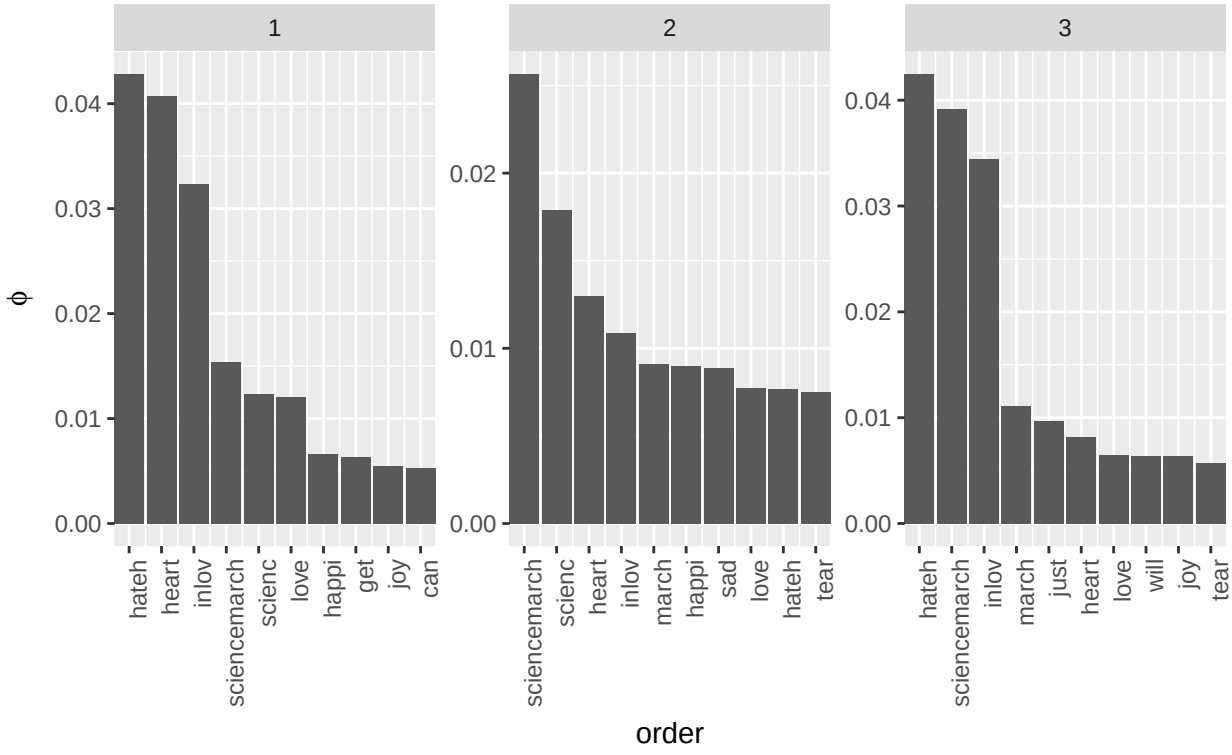


Figure 5: Graphical display of Table 7

4.2.3 LDA with emoji characters as unicode

For the last case, LDA was performed with the data set including emoji characters as unicode. The Table 8 and Figure 6 showed that LDA successfully distinguished the corpus into three different topics. The bar chart illustrated that Topic 1 is related to 'Science March', Topic 2 is related to 'Hate her', and Topic 3 is related to 'In Love'. In order to visualize the probability of a word given topic, a ternary plot was constructed in 3c. The ternary plot showed that four different groups of words: three groups of words were strongly affiliated with one of the topics. These groups of words were clustered at the end of each vertices. The other group of words was located at the center of the triangle which represented words that are not associated with a specific topic. According to this ternary plot, emoji characters had strong connection with a particular topic.

	1.term	1.phi	2.term	2.phi	3.term	3.phi
1	inlov	0.0910	sciencemarch	0.0881	hateh	0.1003
2	😍	0.0302	scienc	0.0322	😂	0.0139
3	love	0.0241	march	0.0182	just	0.0108
4	❤️	0.0163	will	0.0092	🙄	0.0085
5	beauti	0.0073	join	0.0087	get	0.0061
6	just	0.0062	marchforsci	0.0082	like	0.0061
7	new	0.0060	saturday	0.0081	bitch	0.0055
8	💕	0.0049	sign	0.0063	cbb	0.0042
9	happi	0.0045	april	0.0058	one	0.0040
10	inlove...	0.0043	scientist	0.0056	want	0.0039

Table 8: LDA output including emoji characters as unicode

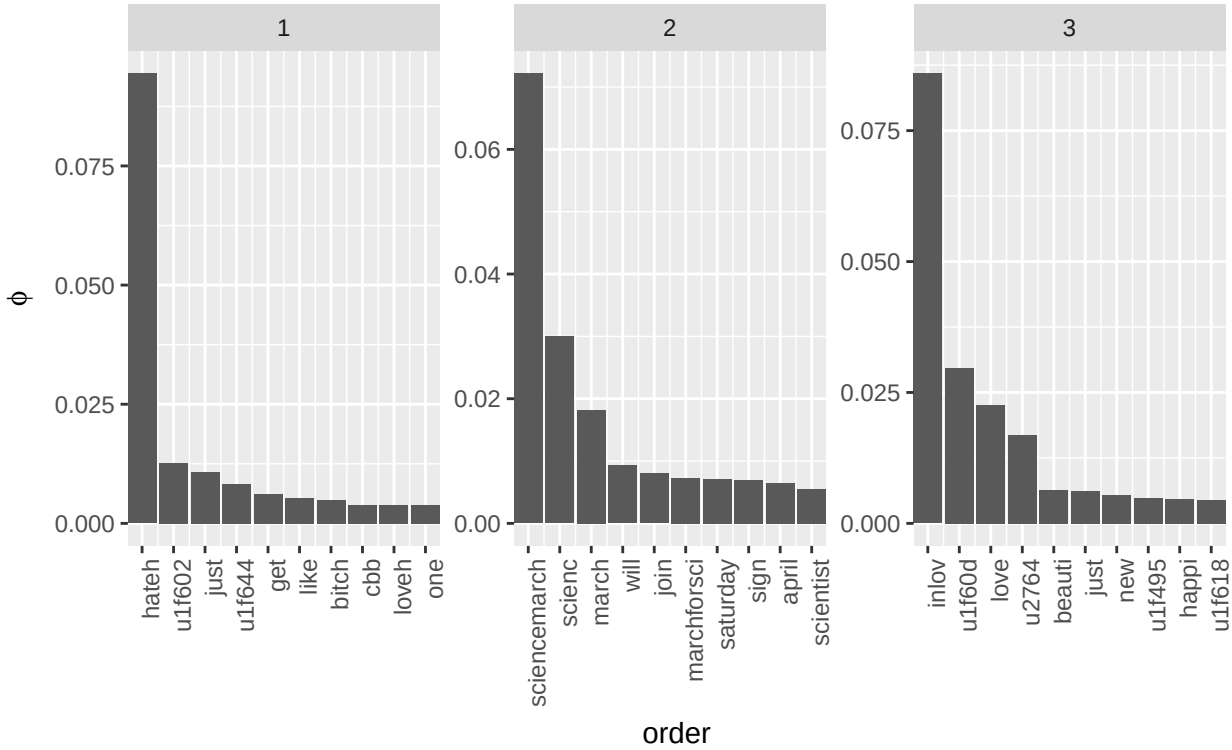


Figure 6: Graphical display of Table 8

5 Conclusion and Discussion

Using Twitter text messages of three different hastags, we examined the performance of LDA considering three different cases with emoji characters. Overall, our results from 3 suggest that LDA with emoji characters embedded in the text data as unicode performed the best. 3c was not only able to assign similar words to each latent topic but also determined terms that were used among all topics such as “just”, “one”, and “like” and displayed at the center of the plot.

One explanation of the outcome is that translating emoji characters into English affects the performance of LDA by introducing new words with sparsity issue or words that could be assigned to multiple different topics. Emoji characters aggregate – otherwise separated – n-gram words into a single character. For example, commonly used Unicode characters in the data set such as ‘U+1F602’, ‘U+1F644’, ‘U+1F60D’, ‘U+1F618’, ‘U+2764’, and ‘U+1F495’ can be translated to English as ‘face with tears of joy’, ‘face with rolling eyes’, ‘smiling face with heart eyes’, ‘face blowing a kiss’, ‘red heart’, and ‘two hearts’ respectively.

The translation approach introduces new words, and even after removing stopwords and stemming, the distribution of words in the data set changes. This approach may increase the number of words with insignificant meanings or it may introduce data sparsity for words that are important for topic modeling. As alluded to previously, the multi-gram translations share words such as ‘face’ for multiple emojis that may not have the same characteristic. Since LDA methods are affected by the frequency of words in the data set, the effect of the translation method as aforementioned above may affect the performance of the LDA as shown in case three. Since only uni-gram-level LDA was examined in this paper, a multi-gram-level LDA with English translated emoji maybe considered as a future research topic.

Moreover, deleting emoji characters will lead to information loss in the text data. Emoji are rich in contextual information, thus deleting the entire emoji characters lead to a greater information loss and consequently affected the output of the LDA. The comparison of the 3b and 3c supports this claim.

In conclusion, this paper provided several insights that have implication for the performance of the LDA considering emoji characters. Considering that emojis are widely used on various SNS platforms and these SNS data are becoming importat as marketing and social studies, our reults suggest that emoji characters should be kept as unicode characters throughout the LDA analysis.