Latent Dirichlet Allocation Models Considering Emojis

Taikgun Song

Abstract

In this research, we compared the performance of <!–a popular topic modeling method–> the Latent Dirichlet Allocation (LDA) by examining the effect of emoji characters. We explored the impact of the emoji characters with respect to LDA by conducting LDA under three different cases: (1) LDA with raw text data, (2) LDA with Emoji characters deleted, and (3) LDA after trasnalting the emoji characters into English. Using Twitter text messages of three different hastags, out analyses revealed that case (1) LDA with raw text data identified the topics the best. Our results highlight that an emoji character aggregates information of n-gram words and the loss of information by deleting the emoji or translating the Emoji into multi-gram English words significantly reduces the LDA performance.

Contents

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.

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.

Let M be the total number of documents in the data set, and N_m be the number of words in the m^{th}

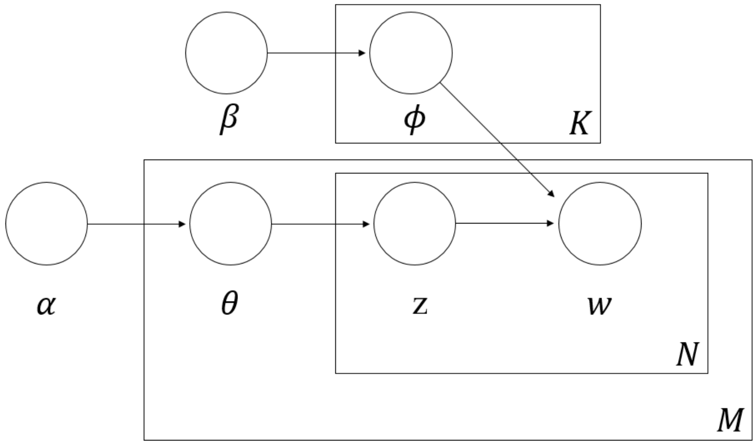


Figure 2: Graphical Model representation of LDA in plate notation.

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 $\boldsymbol{\beta} = \left[\beta_1 \cdots \beta_N\right]$. 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 $\boldsymbol{\alpha} = \left[\alpha_1 \cdots \alpha_K\right]$.

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 $m = 1, \ldots M$ and $\alpha > 0$. For a given topic $z_m = k$, we assume that $w_{mn} \sim Multinomial(\boldsymbol{\phi}_k), n = 1, \ldots, n_m, m = 1, \ldots, M$, where $\boldsymbol{\phi}_k \sim Dirichlet(\boldsymbol{\beta})$, $k = 1, \ldots, K$.

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, $m \in \left\{1, \ldots, M\right\}$ and $n \in \left\{1, \cdots, N_m\right\}$
5. z_{mn}: The topic of the w_{mn}, $z_{mn} \in \left\{1, \ldots, K\right\}$
6. $\boldsymbol{\alpha}$: A vector of prior weights for each topic in a document
   $\boldsymbol{\alpha} = \left[\alpha_1 \cdots \alpha_K\ \right]$
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
   $\boldsymbol{\beta} = \left[\beta_1 \cdots \beta_N\ \right]$
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.

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

Data preparation

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

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

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.

Application

Data Set and exploratory data analysis

Two samples of twitter messages with the following hash-tag #inlove and #hateher 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 (Note that there may be multiple emojis per Twitter message). Top 5 frequently used emojis per hashtag is given in Table 5.

#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

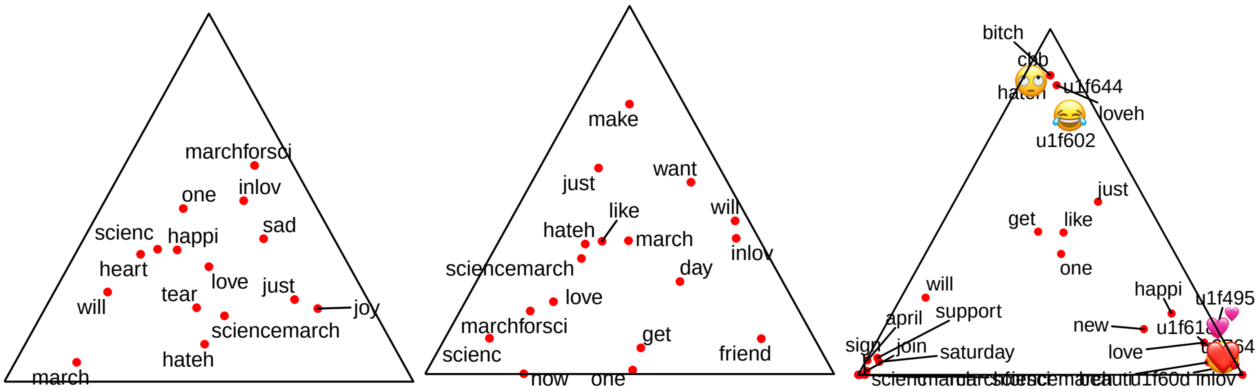
Table 5: Five most popular emoji for each hashtag

It is 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.

Results

LDA was performed on the following three difference cases:

1. LDA on a raw data set
2. LDA on a data set with Unicode removed
3. LDA on a data set with emoji translated to text



(a) Translating All Unicode Characters (b) Excluding All Unicode Characters (c) With Raw Unicode Characters

Figure 3: LDA Output Displayed as a Ternary Plot for each Methods

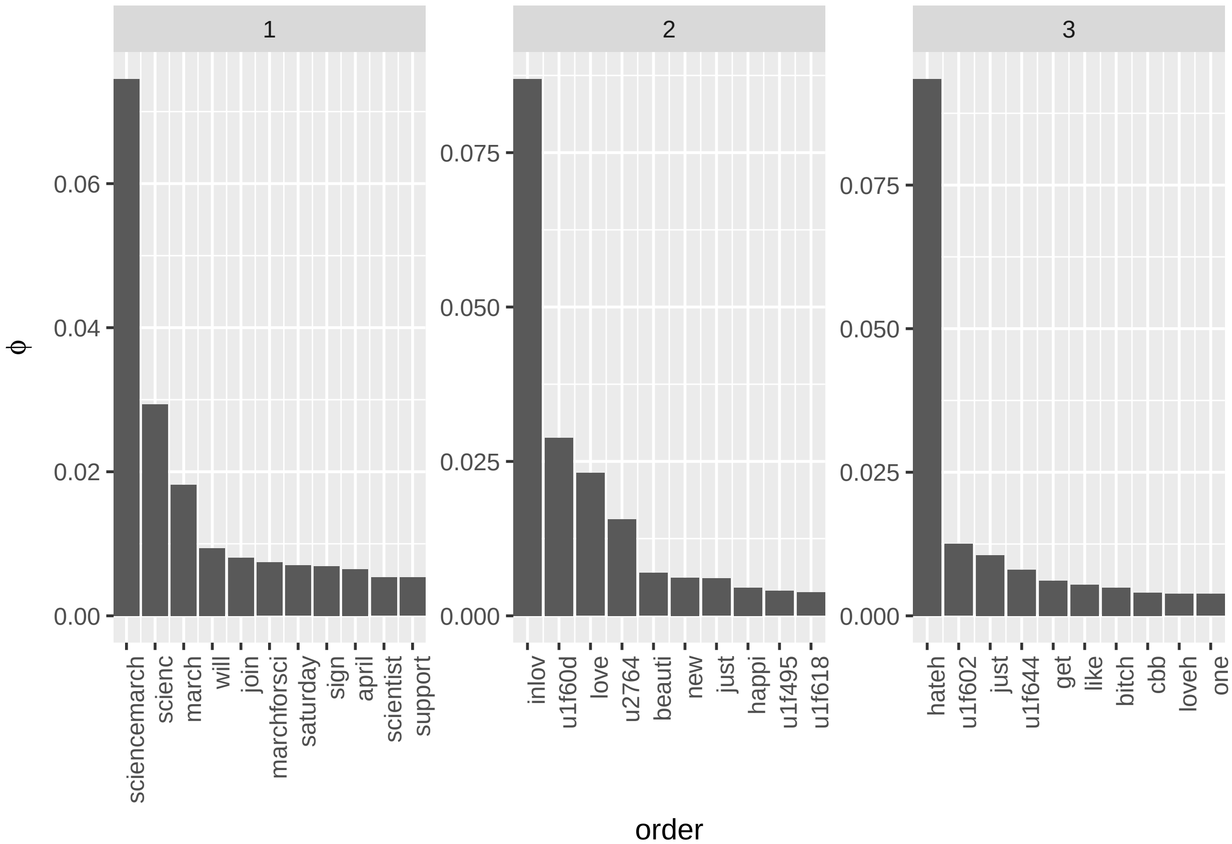
A ternary plot is a triangular graph that displays three variables with respect to its proportion that sum to one. Ternary plots were used to illustrate the LDA ouput of the three different cases listed above. Each corner of the triangle represents the topic determined by the LDA method. The conditional 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.

LDA on a raw data set

The first case was to run LDA on a raw data set. Different number of topic dimensions were tested and the result of 3 topic dimension with 10 terms is provided in **??**. The bar chart in **??** 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.

Table 6: Output LDA with the raw data

Topic 1	Topic 2	Topic 3
sciencemarch	hateh	inlov
scienc	u1f602	u1f60d
march	just	love
will	u1f644	u2764
join	get	beauti
marchforsci	like	new
saturday	bitch	just
sign	cbb	happi
april	loveh	u1f495
support	one	u1f618



LDA without Unicode

In most text mining examples, LDA is performed after removing the Unicode information. For the second case, Unicode characters were removed from the raw text data set before performing LDA. The result of the LDA with three topic dimensions is provided in Table 8. The bar-chart in **??** demonstrated that the LDA method did not distinguish the three topics well. 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 shows that the output from the LDA was located at the center of the ternary plot. This indicates that words had weak connection to any topic dimensions.

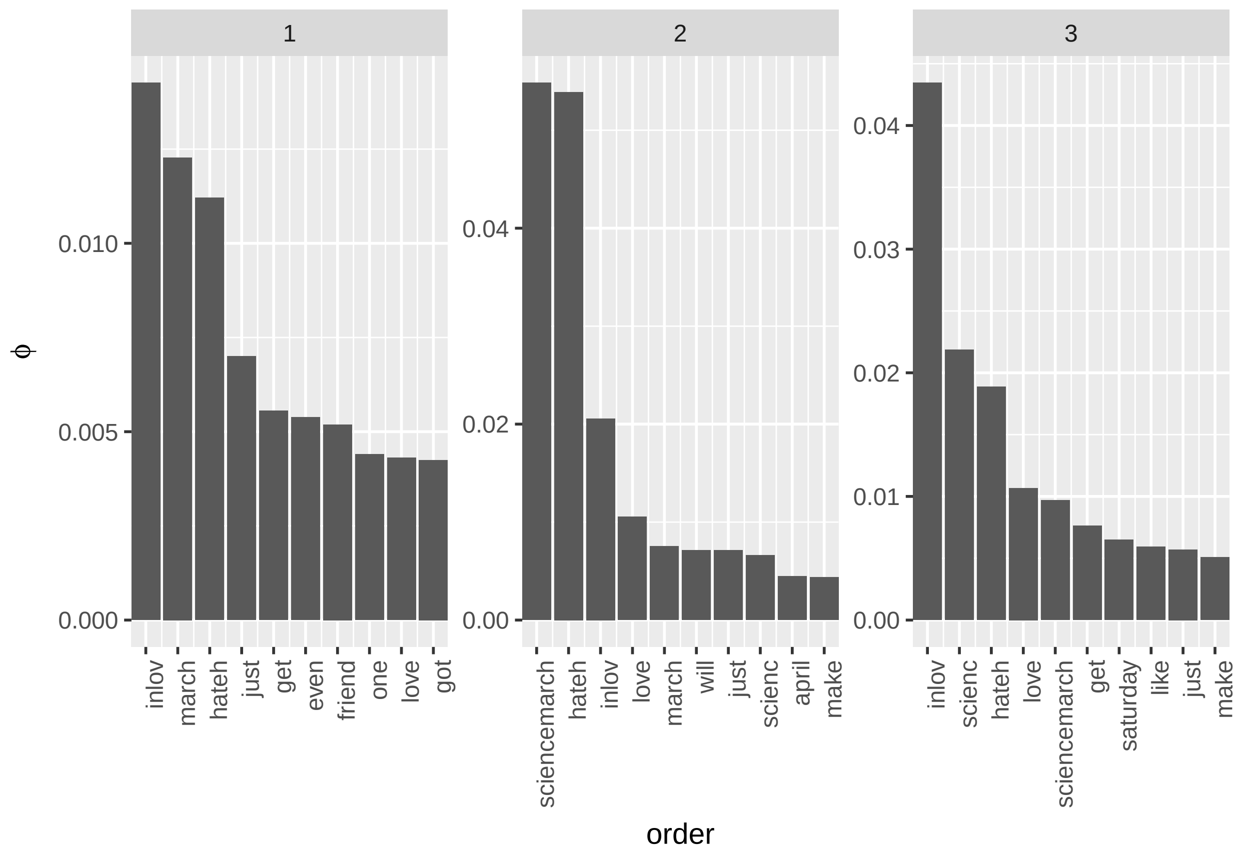
Table 7: Output of LDA with the raw data without the Unicode

Topic 1	Topic 2	Topic 3
hateh	hateh	inlov
sciencemarch	inlov	hateh
scienc	sciencemarch	sciencemarch
love	just	will
get	make	get

Table 8: Word prob. given topic

1.term	1.phi	2.term	2.phi	3.term	3.phi
hateh	0.0403	hateh	0.0406	inlov	0.0455

1.term	1.phi	2.term	2.phi	3.term	3.phi
sciencemarch	0.0375	inlov	0.0354	hateh	0.0186
scienc	0.0258	sciencemarch	0.0311	sciencemarch	0.0172
love	0.0158	just	0.0123	will	0.0081
get	0.0060	make	0.0094	get	0.0068
march	0.0059	march	0.0084	march	0.0058
marchforsci	0.0053	will	0.0075	love	0.0055
one	0.0051	love	0.0062	one	0.0052
now	0.0049	want	0.0053	friend	0.0050
just	0.0048	like	0.0052	day	0.0045



LDA with name translated

The last case was to perform LDA after translating the Unicode emoji characters in English. The output of LDA with translation is provided in **??**. The bar-chart in **??** 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.

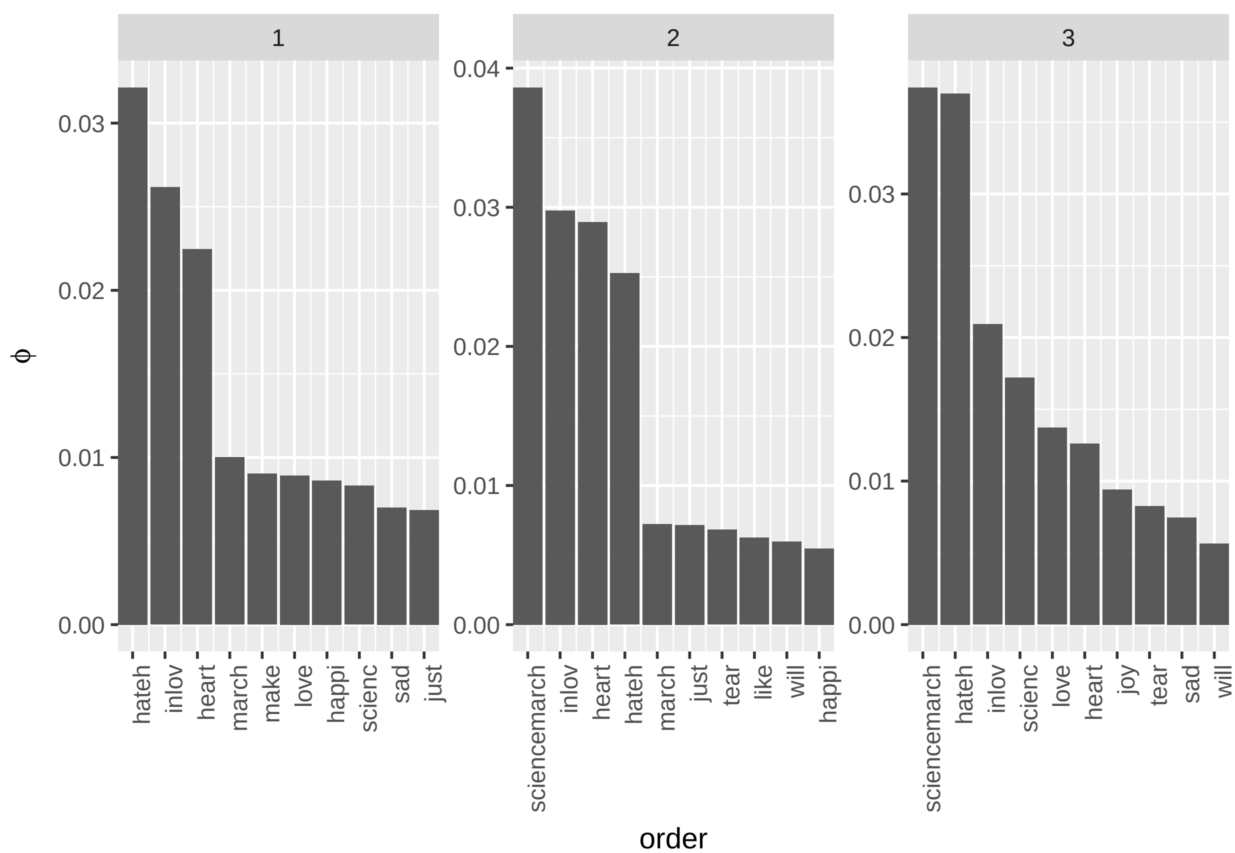
Table 9: Output of LDA with translated Unicode

Topic 1	Topic 2	Topic 3
hateh	inlov	hateh

Topic 1	Topic 2	Topic 3
heart	heart	sciencemarch
sciencemarch	sciencemarch	inlov
march	scienc	just
scienc	hateh	joy

Table 10: Word prob. given topic

1.term	1.phi	2.term	2.phi	3.term	3.phi
hateh	0.0428	inlov	0.0435	hateh	0.0408
heart	0.0296	heart	0.0253	sciencemarch	0.0342
sciencemarch	0.0282	sciencemarch	0.0162	inlov	0.0230
march	0.0158	scienc	0.0129	just	0.0102
scienc	0.0129	hateh	0.0111	joy	0.0101
inlov	0.0100	love	0.0093	heart	0.0084
will	0.0090	happi	0.0074	love	0.0083
love	0.0083	sad	0.0068	sad	0.0062
tear	0.0068	marchforsci	0.0050	tear	0.0058
happi	0.0066	one	0.0050	scienc	0.0051



Conclusion and Discussion

Using Twitter text messages of three different hastags, we examined the performance of LDA with respect to emoji characters. Overall, our results suggest that LDA with raw emoji characters embedded in the text data performed the best. While the LDA method when all emojis were deleted and the LDA method when all emojis were translated genertated ternary plots 3a and 3b where words were clusterd at the center of the graphs. This indicate that the LDA method was not able to assign words into three unknown topics. On the other hand, 3c showed that three clusters of words were plotted at each end point of the triangle for method when raw unicode characters were used for the LDA method. These plots indicate that LDA performed the best when the raw unicode characters were used for the analysis.

One explanation of the outcome is that deleting emoji characters will lead to information loss in the text data. Emoji are rich in information, thus deleting the entire character led to great information loss and consequently affected the output of the LDA in case two. The comparison of 3b and 3c supports this claim.

Moreover, emoji characters aggregates – 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. Although the meaning of each emoji characters are different, their multi-gram translations share words such as 'face', and 'heart'. Also, it produces a few meaningless as a byproduct of translation. Since LDA methods are affected by the frequency of words in the data set, the effect of the translation method as alluded to above may affect the performance of the LDA as shown in case three.