Latent Dirichlet Allocation Models Considering Emojis

Taikgun Song

0.0.1 abstract

XXX write later XXX

Contents

1 Introduction

Text data contains valuable insights that may be useful for content recommendation, customer care service, social media analysis, and *et cetera*. However, these information are usually hidden underneath the plain text. Topic modeling is a text-mining method that extracts information from a text data by identifying latent semantic structures in the text body. One of the most widely used as a topic modeling method is the Latent Dirichlet Allocation(LDA). LDA is a popular hierarchical Bayesian model which assumes that each of the documents in a collection consist of a mixture of topics, and these topics are reponsible for the establishment of words in each document. Topics, however, are the latent part of the document set and one can only observe words collected into documents. LDA exploits 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. Therefore, evaluating this primary 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 propose 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 have three main benefits. First, it may reduce the systematic problem of LDA with data sparcity. All emoji characters have name and keywords associated with the contextual meaning that it conveys. By translating emoji characters into its English name or related keywords will increase the observation, and thus lead to better LDA results. Second, each emoji character has a couple of pre-determined topic dimension set by the official organization. This information could be used as an auxiliary information during the topic matching process. Lastly, 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.

What do we want to learn from the messages? This is where the problem statement goes.

2   LDA

Let w_{mn} be the nth word in the mth 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 mth document

3. K: The total number of topics in the data set

4. w_{mn}: n^{th} word in document m, $m \in \{1, \ldots, M\}$ and $n \in \{1, \cdots, N_m\}$

5. z_{mn}: The topic of the w_{mn}, $z_{mn} \in \{1, \ldots, K\}$

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 distirbution 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. $w_{mn} \sim Multinomial(\phi_{z_{mn}})$

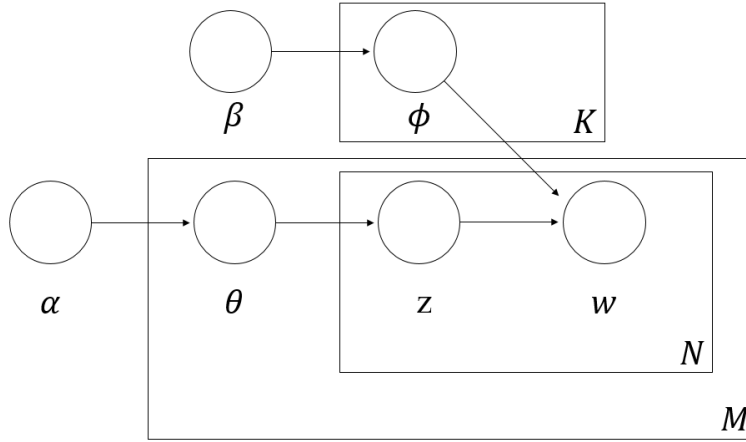The graphical display of LDA is given in Figure 2.

Figure 2: Graphical Model representation of LDA

Then, the total probability of the model is

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

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

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

On the application perspective, solving the posterior distribution is the key which is given as the following equation.

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

However, this posterior distribution is intractable for exact inference.

intractable. LDA package uses Gibbs sampling approach. explain this in more details.

3 Data preparation

data comes with text format. need data preperation.

stop words, stemming, ... should go into a separate section called '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 method vulnerable when meaningless words such as function/structural words are present in the data set with high frequency. Thus, any group of non-informative words including the function/structural words should be filtered out before doing an analysis, and this group of words are called the `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 work done in the paper, the `tm` package in `R` was used to delete the stop words.

give some examples following the tweets or again go back to the xkcd example

	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 a root word "stem". Words with same meaning but different in forms contribute to data sparsity, reducing the performance of the LDA method. The `stemming` procedure cuts inflectional forms of a word to its root form eventually increasing the frequency of word observations.

The stemming process has two disadvantages. First, there are 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 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.

include a couple of examples

	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-grams and just generally features of documents feature extraction. n-grams is just one of them and I have used uni-gram. justify why.

3.3 n-gram

n-gram is a neighboring sequence of n items from a collection of text data set. Although this item could be anything from phonemes or syllables to letters and words based on the application. 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

For this paper, a single word or 1-gram, was considered as the base unit of the item.

4 Application

4.1 Data Set and exploratory data analysis

more info on the data: use dates - should we wrap this into a shiny app down the road?
Shiny had a problem with instant web scraps last year. I am not certain if that problem is fixed now.

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.

4.2 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

4.3 LDA on a raw data set

The second case was to run LDA on a raw data set. Stemming and stop word deletion were performed. Different number of

Table 6: Output LDA with the raw data

Topic 1	Topic 2	Topic 3
inlov	hateh	sciencemarch
hateh	sciencemarch	inlov
u1f60d	scienc	hateh
love	like	scienc
u1f602	march	march
sciencemarch	sign	love
new	saturday	u2764
u2764	will	just
just	make	get
scienc	just	u1f60d

Table 7: Word prob. given topic

| 1.term	1.phi	2.term	2.phi	3.term	3.phi
inlov	0.05267	hateh	0.05424	sciencemarch	0.04419
hateh	0.02686	sciencemarch	0.0342	inlov	0.03118
u1f60d	0.02276	scienc	0.01265	hateh	0.01756
love	0.01541	like	0.008221	scienc	0.01608
u1f602	0.01465	march	0.007597	march	0.01083
sciencemarch	0.01297	sign	0.007225	love	0.01083
new	0.009023	saturday	0.006626	u2764	0.009843
u2764	0.007257	will	0.006558	just	0.007875
just	0.006742	make	0.005885	get	0.006345
scienc	0.005824	just	0.004944	u1f60d	0.005952
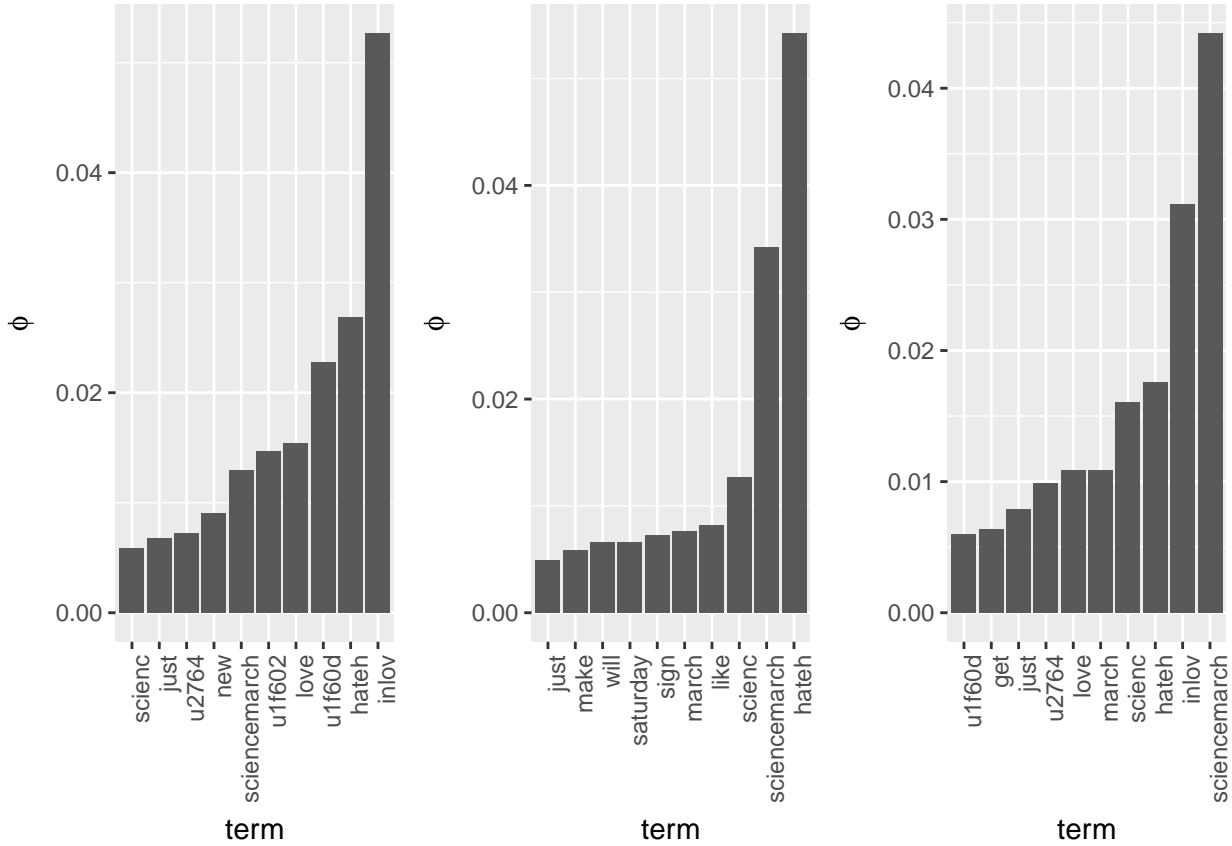
4.4 LDA without Unicode

In most text mining examples, LDA is performed after removing the Unicode information. For the first case, therefore, Unicode characters were removed from the raw text data set. Then, the standard procedure of stemming and stop word deletion was performed to enhance the accuracy of LDA. `tm` package was used to conduct the above procedure.

Table 8: Output of LDA with the raw data without the Unicode

Topic 1	Topic 2	Topic 3
sciencemarch	inlov	sciencemarch
hateh	hateh	hateh
love	love	inlov
scienc	scienc	scienc
will	make	march

Table 9: Word prob. given topic

| 1.term	1.phi	2.term	2.phi	3.term	3.phi
sciencemarch	0.02292	inlov	0.04221	sciencemarch	0.05955
hateh	0.01418	hateh	0.03178	hateh	0.05161
love	0.01384	love	0.0216	inlov	0.03657
scienc	0.008893	scienc	0.0124	scienc	0.01587
will	0.008087	make	0.01155	march	0.008755
one	0.007756	march	0.007855	sat	0.008243
need	0.005949	want	0.00617	get	0.005635
march	0.005753	like	0.006129	like	0.004787
say	0.005431	sciencemarch	0.005831	dont	0.004741
new	0.005303	abo	0.005344	abo	0.004685

7
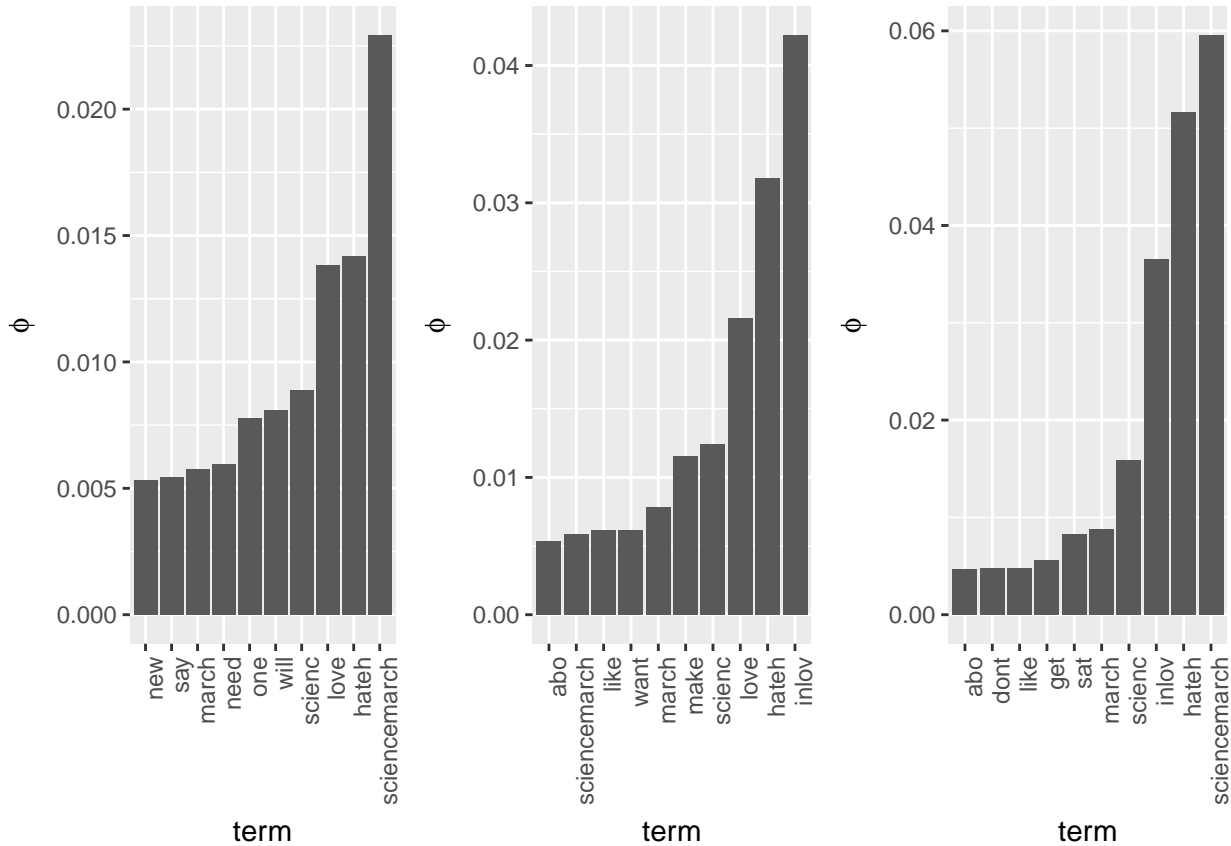
4.5 LDA with name translated

The last case was to perform LDA after translating the Unicode emoji characters in English. `unicode` package was used to match the Unicode to its name. Then the standard process of stemming and deletion of stop words where performed.

Table 10: Output of LDA with translated Unicode

Topic 1	Topic 2	Topic 3
hateh	inlov	sciencemarch
inlov	hateh	heart
heart	scienc	hateh
happi	like	inlov
get	join	scienc

Table 11: Word prob. given topic

1.term	1.phi	2.term	2.phi	3.term	3.phi
hateh	0.04222	inlov	0.03946	sciencemarch	0.05751
inlov	0.0286	hateh	0.03407	heart	0.02817
heart	0.02469	scienc	0.012	hateh	0.02772
happi	0.01451	like	0.01128	inlov	0.02122
get	0.01164	join	0.00666	scienc	0.01482
love	0.009732	last	0.006403	love	0.01074
march	0.006691	love	0.005753	just	0.0104
scienc	0.005869	make	0.005513	joy	0.009451
will	0.005788	just	0.005287	march	0.007922
tear	0.005767	sciencemarch	0.004866	sad	0.006489
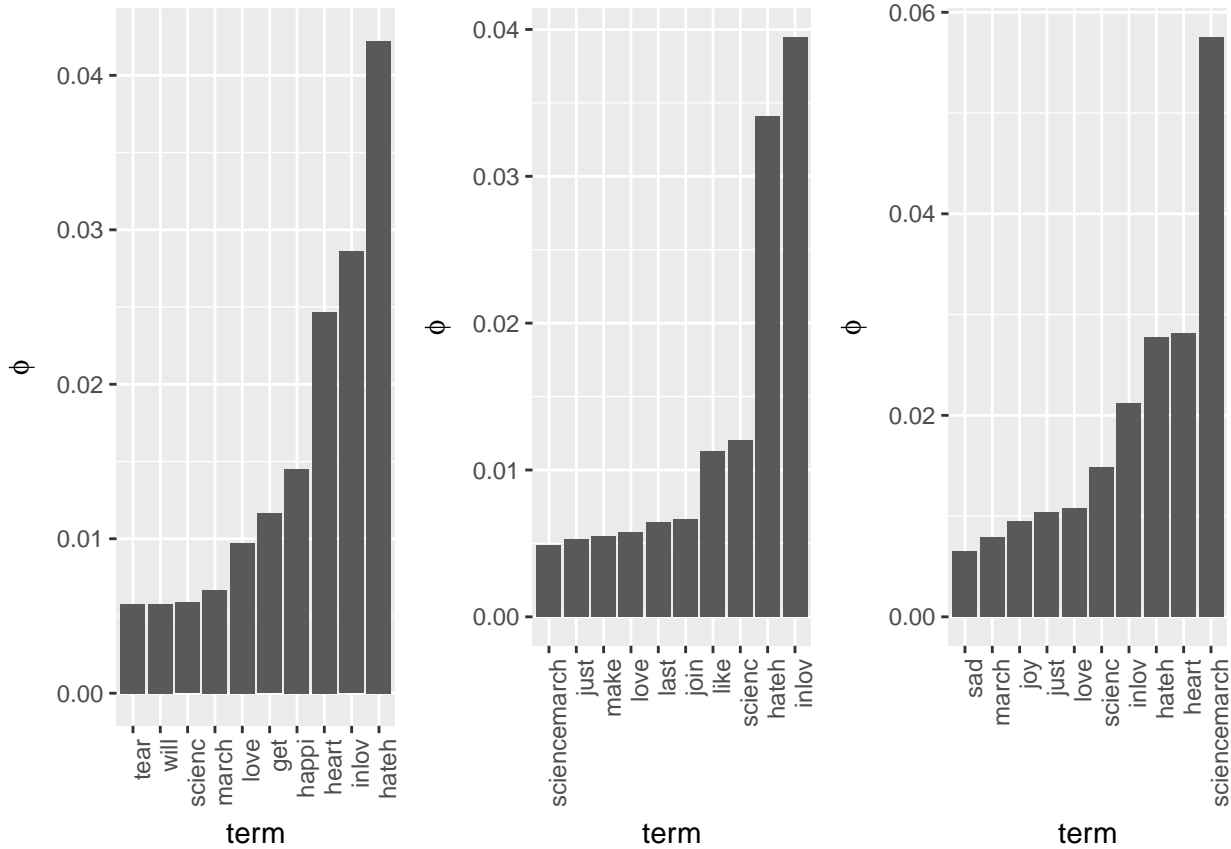
5 Conclusion

As the result of the exploratory analysis indicates, user-generated-contents may contain Unicode emoji characters. These emoji characters sometimes carry mixture of condensed information that is difficult to express in words. The result of the output from the LDA indicates that words such as "heart" that would have been neglected using the traditional method may be saved when the Unicode characters are translated into meanings.

6 Appendix

7 `emoji` package in R

Plan to change this part after posting the emoji package on CRAN

7.1 Description of the emoji package

The `emoji` package contains information of the emoji v5.0 from its official publisher the Unicode Consortium. The illustration of the web page is shown in Figure 3.

Figure 3: Glimpse of the table of emoji on the Unicode.org website

The data set `emoji` in the `emoji` package contains 8 variables:

uni_no: Official number of emojis
uni_code: Formal Unicode of emojis
uni_name: Official name of emojis
cat1: Official category of emojis
cat2: Official sub-category of emojis from cat1
cat3: Official sub-category of emojis from cat2
uni_keyws: Official keyword(s) of emojis
uni_png: Image of emojis in PNG format represented in a matrix format

The package has a function `emoji_info_table` that summarizes all emoji and their information used in a single character string.

7.2 Scoring of Sentiment

The characteristic of emoji (effectively delivers feelings and moods), naturally leads text mining with emoji to sentiment analysis. `tidytext` package in R has three general purpose lexicon sets. The `AFINN` score words from -5 to 5 scale, `bing` assigns words in binary category(positive and negative), and `nrc`assigns words with more categories.

8 More work

0. Technical details The `tm`, `topicmodels`, `emoji`, `tidytext`, and `tidyverse` package in `R` was written to help the above analysis.
1. Check Stemming - scienc vs. science
2. Check output again. Also, a check aggregation of short messages to avoid data sparsity.
3. LDA explanation
4. Description of the emoji package