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.

XXX Should the packages used to run example be introduced here with brief steps? XXX the packages should go to the back into a 'technical details'.

The `tm`, `topicmodels`, `emoji`, `tidytext`, and `tidyverse` package in `R` was written to help the above analysis.

2 LDA

Assume that there is a text data consisting M number of documents. Further assume that there are K number of topics in the data set. In a text data, words are the only observable variable, and other variables such as the topic variable is hidden. LDA assumes that documents are represented as random mixtures over latent topic, and each topic is realized in terms of words.

Let w_{mn} be the n^{th} word observed in the m^{th} document with topic z_{mn}. The topic z_{mn} is drawn from a distribution of topics in document m, denoted as θ_m. Moreover, since each topic is characterized by words, word w_{mn} is drawn from a distribution of topic z_{mn}, denoted as $\phi_{z_{mn}}$. LDA assumes that θ follows a Dirischlet distribution with parameter α, and ϕ follows a Dirischlet distribution with parameter β.

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, $m \in \{1, \ldots, M\}$ 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
7. θ_m: The distribution of topics in document m
   $\theta_m \sim Dir(\boldsymbol{\alpha})$
8. $\boldsymbol{\beta}$: A vector of prior weights for each word in a topic
9. ϕ_z: The distirbution of words in topic z
   $\phi_z \sim Dir(\boldsymbol{\beta})$
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.

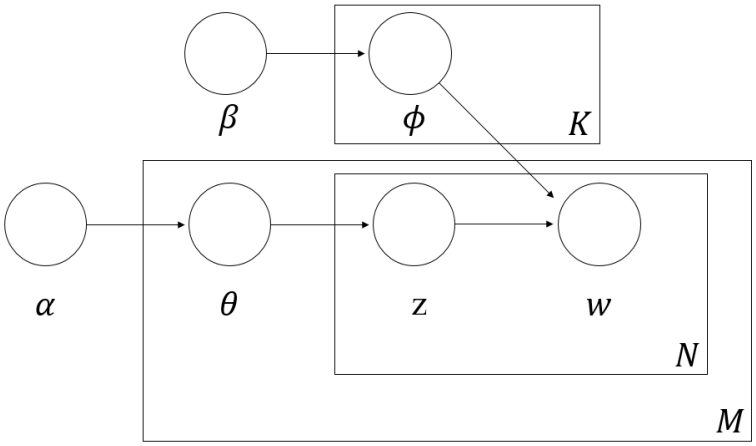


Figure 2: Graphical Model representation of LDA

Then, the total probability 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:

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

where

2.1 LDA Equation goes here

As indicated in the above section, LDA assumes that documents are represented as random mixtures over latent topics and each topic is characterized by a distribution over words. Therefore, the frequency of each word influence the outcome of the LDA.

stop words, stemming, ... should go into a separate section called 'Data preparation'

2.2 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 freqeuncy. 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 stop words.

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

2.3 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 sparcity, 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.

XXX Explain why we cannot trace back to the original form XXX there's different ways to resolve that - most times we use the most frequent word/version for this stem.

include a couple of examples

The `tm` package is again used for the stemming process and its code is given as the following.

n-grams and just generally features of documents

3 Application

3.1 Data Set and exploratory data analysis

more info on the data: use dates - should we wrap this into a shiny app down the road?

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 1. 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 1: 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 2.

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.

3.2 Results

LDA was performed on the following three difference cases:

#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 2: Five most popular emoji for each hashtag

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

3.3 LDA on a raw data set

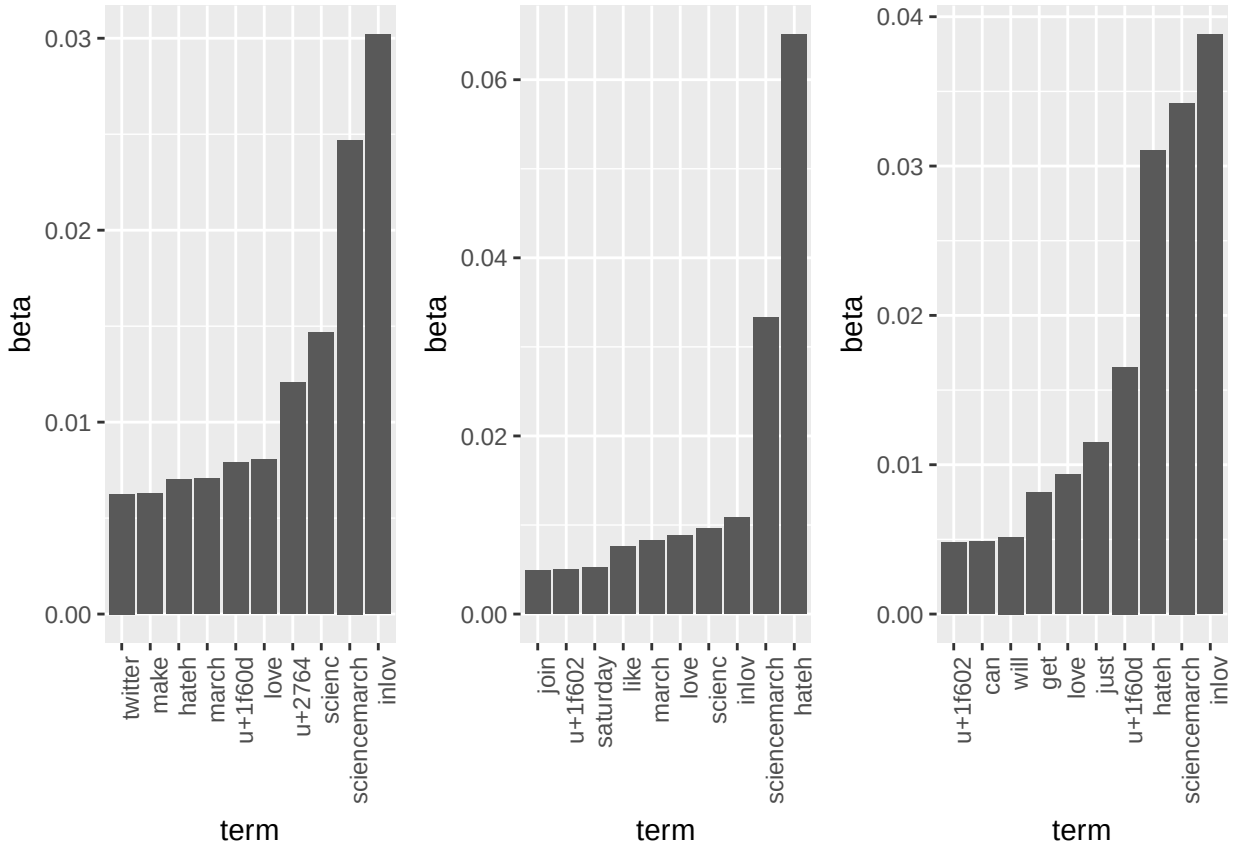
The second case was to run LDA on a raw data set. Stemming and stop word deletion were performed. Different number of topic dimensions were tested and the result of 4 topic dimension with 10 terms are provided in Table 3. Describe the output.

Table 3: Output LDA with the raw data

Topic 1	Topic 2	Topic 3
inlov	hateh	inlov
sciencemarch	sciencemarch	sciencemarch
scienc	inlov	hateh
u+2764	scienc	u+1f60d
love	love	just
u+1f60d	march	love
march	like	get
hateh	saturday	will
make	u+1f602	can
twitter	join	u+1f602

Table 4: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
inlov	0.03019	hateh	0.06507	inlov	0.0388
sciencemarch	0.0247	sciencemarch	0.03333	sciencemarch	0.03422
scienc	0.01469	inlov	0.01083	hateh	0.03106
u+2764	0.01205	scienc	0.009612	u+1f60d	0.01655
love	0.008046	love	0.008841	just	0.01151
u+1f60d	0.007876	march	0.008305	love	0.009341
march	0.007083	like	0.00755	get	0.008127
hateh	0.00703	saturday	0.005191	will	0.005163
make	0.00626	u+1f602	0.005049	can	0.00488
twitter	0.006255	join	0.004926	u+1f602	0.004806



3.4 LDA without Unicode

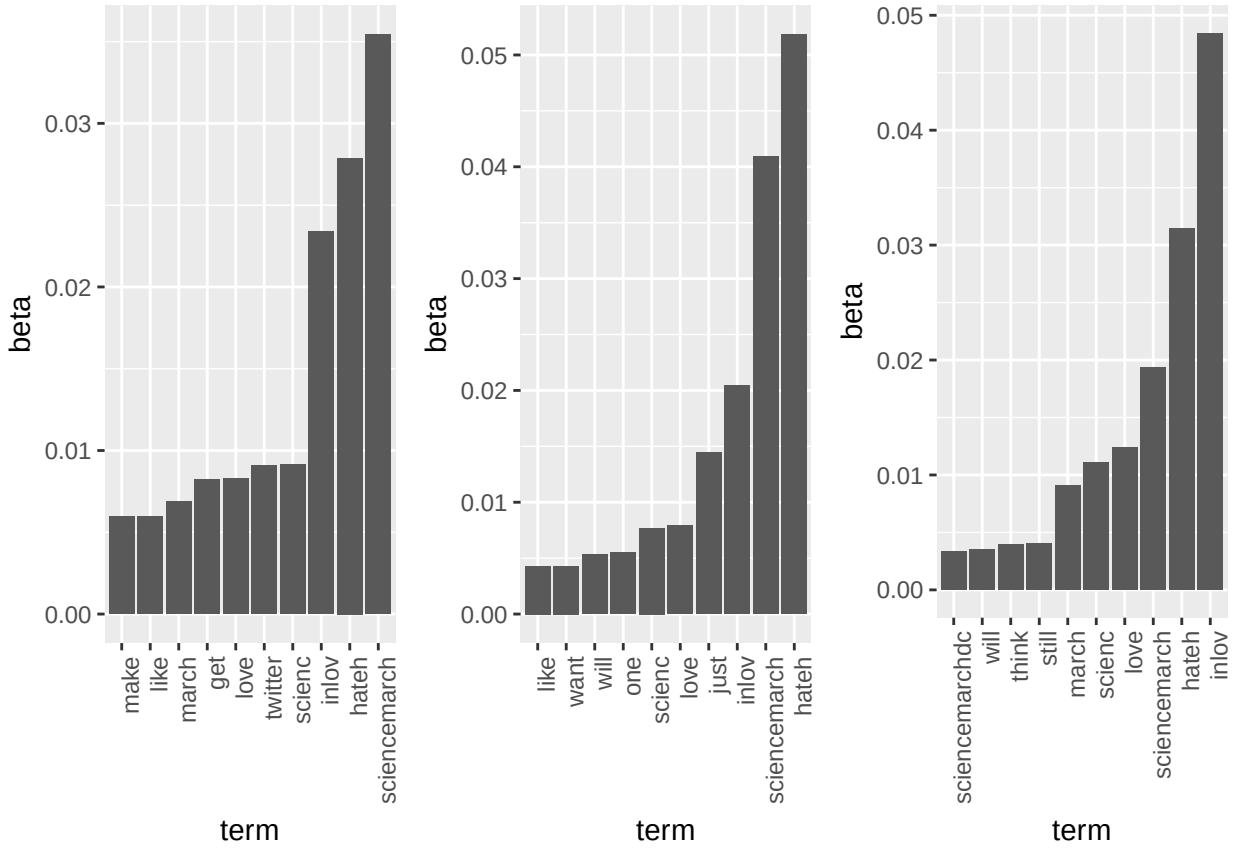
In most text mining examples, LDA is performed after removing the Unicode information. For the first case, therefore, Unicode characters were removed from the raw text data set. Then, the standard procedure of stemming and stop word deletion was performed to enhance the accuracy of LDA. `tm` package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

Table 5: Output of LDA with the raw data without the Unicode

Topic 1	Topic 2	Topic 3
sciencemarch	hateh	inlov
hateh	sciencemarch	hateh
inlov	inlov	sciencemarch
scienc	just	love
twitter	love	scienc

Table 6: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
sciencemarch	0.03543	hateh	0.05182	inlov	0.04842
hateh	0.02788	sciencemarch	0.04095	hateh	0.03147
inlov	0.02337	inlov	0.02041	sciencemarch	0.01934
scienc	0.009166	just	0.01442	love	0.01242
twitter	0.009092	love	0.007895	scienc	0.01112
love	0.008314	scienc	0.007692	march	0.009077
get	0.008244	one	0.005514	still	0.004012
march	0.006904	will	0.005324	think	0.003946
like	0.005988	want	0.004298	will	0.00354
make	0.005956	like	0.004295	sciencemarchdc	0.003326



3.5 LDA with name translated

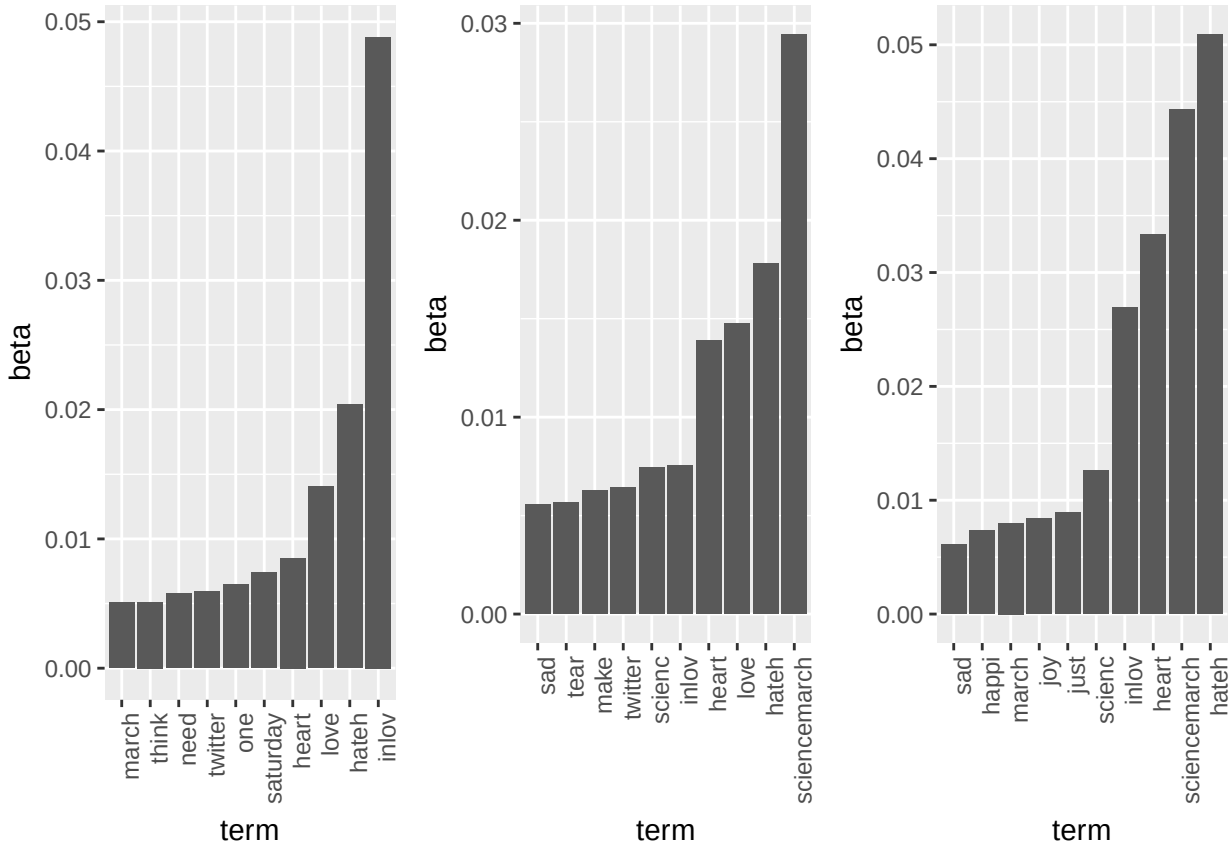
The last case was to perform LDA after translating the Unicode emoji characters in English. `unicode` package was used to match the Unicode to its name. Then the standard process of stemming and deletion of stop words where performed.

Table 7: Output of LDA with translated Unicode

Topic 1	Topic 2	Topic 3
inlov	sciencemarch	hateh
hateh	hateh	sciencemarch
love	love	heart
heart	heart	inlov
saturday	inlov	scienc

Table 8: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
inlov	0.04881	sciencemarch	0.02943	hateh	0.0509
hateh	0.02042	hateh	0.0178	sciencemarch	0.04435
love	0.01407	love	0.01476	heart	0.03331
heart	0.00852	heart	0.0139	inlov	0.02689
saturday	0.00741	inlov	0.007529	scienc	0.0126
one	0.006462	scienc	0.007442	just	0.008891
twitter	0.005932	twitter	0.006417	joy	0.008362
need	0.005768	make	0.006276	march	0.007998
think	0.005129	tear	0.00568	happi	0.007372
march	0.005085	sad	0.005549	sad	0.006111



4 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.

5 Appendix

6 `emoji` package in R

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

6.1 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.

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

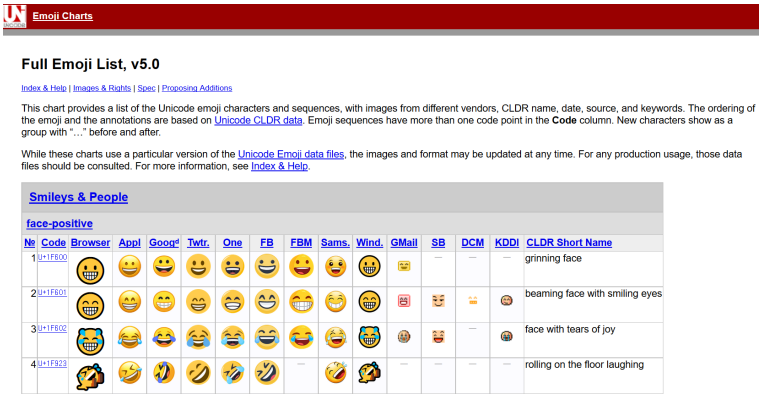


Figure 3: Glimpse of the table of emoji on the Unicode.org website

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.

6.2 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.

Table 9: Example of the emoji package

uni_code	count	name	score	categories	categories2
U+1F469	1	woman	neutral	smileys_&_people, person	female, woman
U+1F495	1	two hearts	positive	smileys_&_people, emotion	love, positive expression
U+1F60F	1	smirking face	neutral	smileys_&_people, face, neutral	expression, face, smirk

7 More work

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