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

flesh this out next. Start with mathematical definitions of the data.

LDA is a popular method to infer semantics to model a document as a mixture of latent topics.

LDA is a topic modeling method that allows words observed in documents to be explained by unobserved topics and that each word's creation is attributable to one of the document's topics.

LDA is a topic modeling method that allows words observed in documents to be explained by unobserved topics and that each word's creation is attributable to one of the document's topics.

LDA is based on the two following principles:

1. Every document is a mixture of topics
2. Every topic is a mixture of words

To illustrate, a news paper document may contain several topics such as "politics", "economy", "spots", "entertainment", and etc. For a given topic "politics", common words may be "government", "trump", "president", "congress", and etc.

LDA assumes that the probability of documents are random mixture over unseen topics, and document i having topic k follows a Dirichlet distribution with some parameter α. That is, if the probability of document i having topic k is denoted as $\theta_{i,k}$, then $\theta_i \sim Dir(\alpha)$. The second assumption says each topic is a mixture of words, and that the distribution of n^{th} word will follow a multinomial distribution conditioned on the topic z. The probability of word given a topic is denoted as β. Then β has a Dirichlet distribution with parameter η.

1. $\theta_i \sim Dir(\alpha), i = 1, \dots, M$
2. $\theta_{i,k}$ is the probability that document $i \in \{1, \dots, M\}$ has topic $k \in \{1, \dots, K\}$.
3. z is word's topic drawn from a Multinomial distribution with parameter θ, i.e. $z \sim Multi(\theta)$
4. $\beta_k \sim Dir(\eta), k = 1, \dots, K$
5. $\beta_{k,v}$ is the probability of word $v \in \{1, \dots, V\}$ in topic $k \in \{1, \dots, K\}$
6. w is a word drawn from a Multinomial distribution with parameter Z and β, i.e., $w \sim Multi(z, \beta)$.

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

Graphical display of LDA is given in Figure 2.

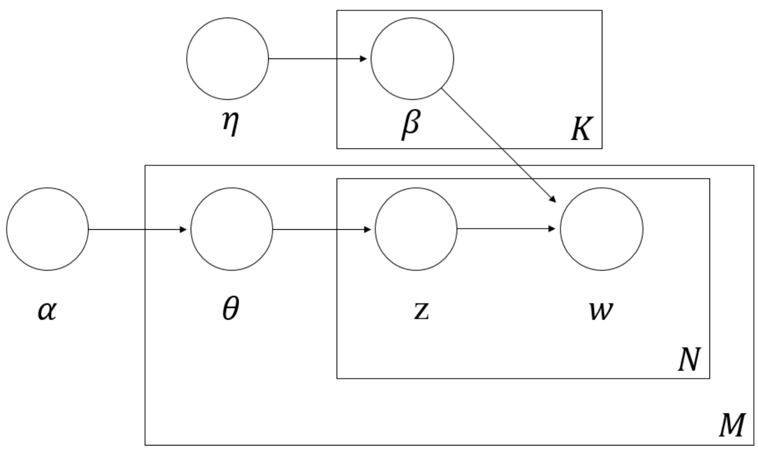


Figure 2: Graphical Model representation of LDA

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.

#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

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:

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

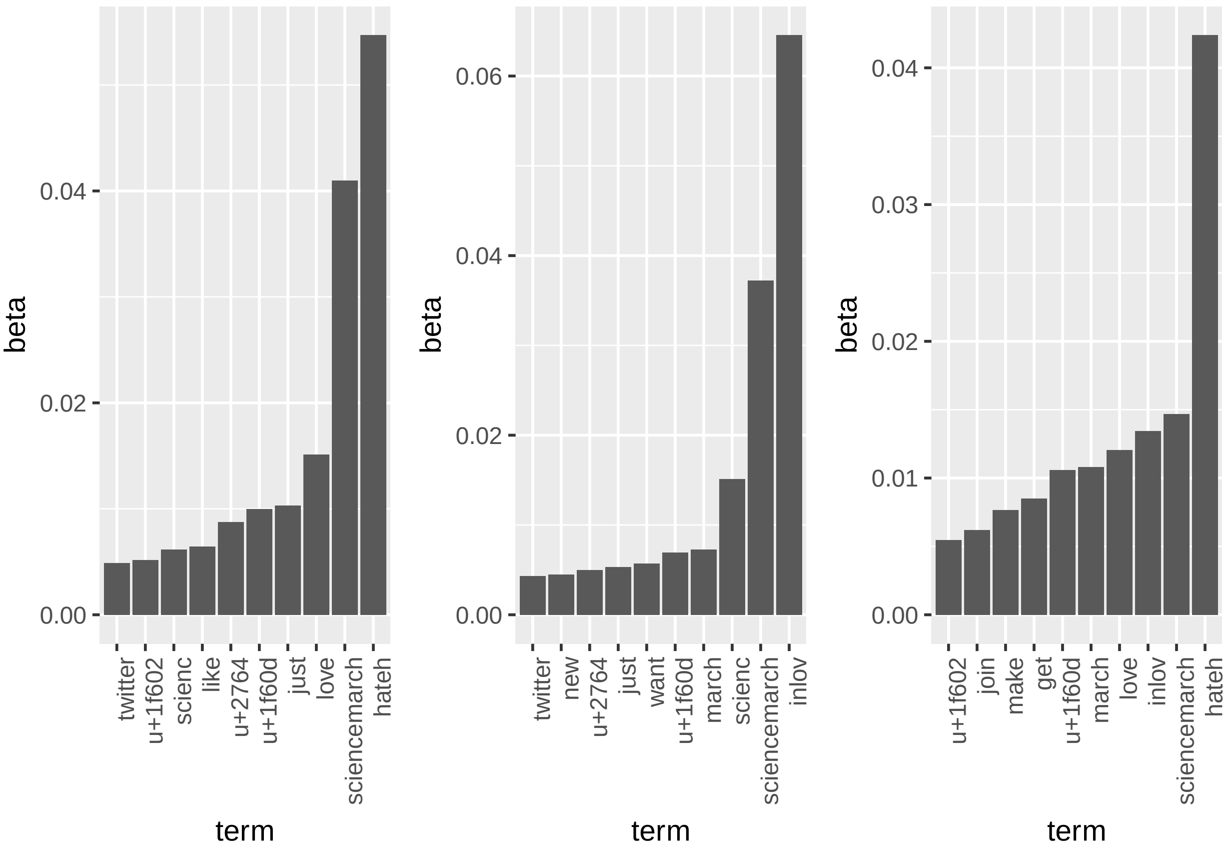
The second case was to run LDA on a raw data set. Stemming and stop word deletion were performed. Different number of topic dimensions were tested and the result of 4 topic dimension with 10 terms are provided in Table 3. Describe the output.

Table 3: Output LDA with the raw data

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

Table 4: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
hateh	0.05469	inlov	0.06454	hateh	0.04238
sciencemarch	0.04098	sciencemarch	0.03724	sciencemarch	0.01469
love	0.01513	scienc	0.01512	inlov	0.01343
just	0.0103	march	0.007262	love	0.01204
u+1f60d	0.009966	u+1f60d	0.006925	march	0.01082
u+2764	0.008765	want	0.005687	u+1f60d	0.01059
like	0.00644	just	0.005299	get	0.008507
scienc	0.00614	u+2764	0.00501	make	0.007673
u+1f602	0.005161	new	0.004499	join	0.006212
twitter	0.004869	twitter	0.004336	u+1f602	0.005466



3.4 LDA without Unicode

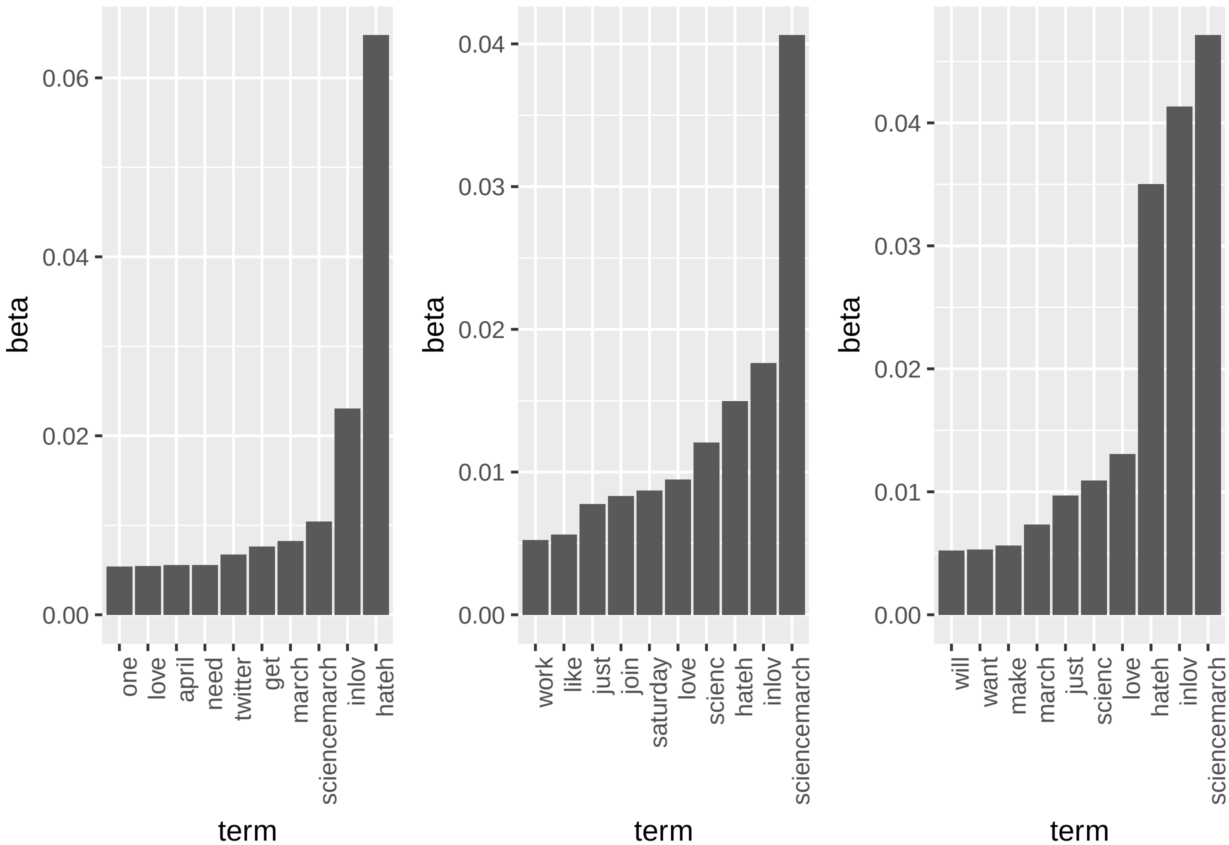
In most text mining examples, LDA is performed after removing the Unicode information. For the first case, therefore, Unicode characters were removed from the raw text data set. Then, the standard procedure of stemming and stop word deletion was performed to enhance the accuracy of LDA. `tm` package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

Table 6: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
hateh	0.06476	sciencemarch	0.0406	sciencemarch	0.04712
inlov	0.02304	inlov	0.01762	inlov	0.04132
sciencemarch	0.01043	hateh	0.01497	hateh	0.03503
march	0.008235	scienc	0.01206	love	0.01305
get	0.007617	love	0.009464	scienc	0.01093
twitter	0.006704	saturday	0.008702	just	0.00969
need	0.005567	join	0.008318	march	0.007335
april	0.005559	just	0.007775	make	0.005632
love	0.005454	like	0.005632	want	0.005288
one	0.005414	work	0.00522	will	0.005214



3.5 LDA with name translated

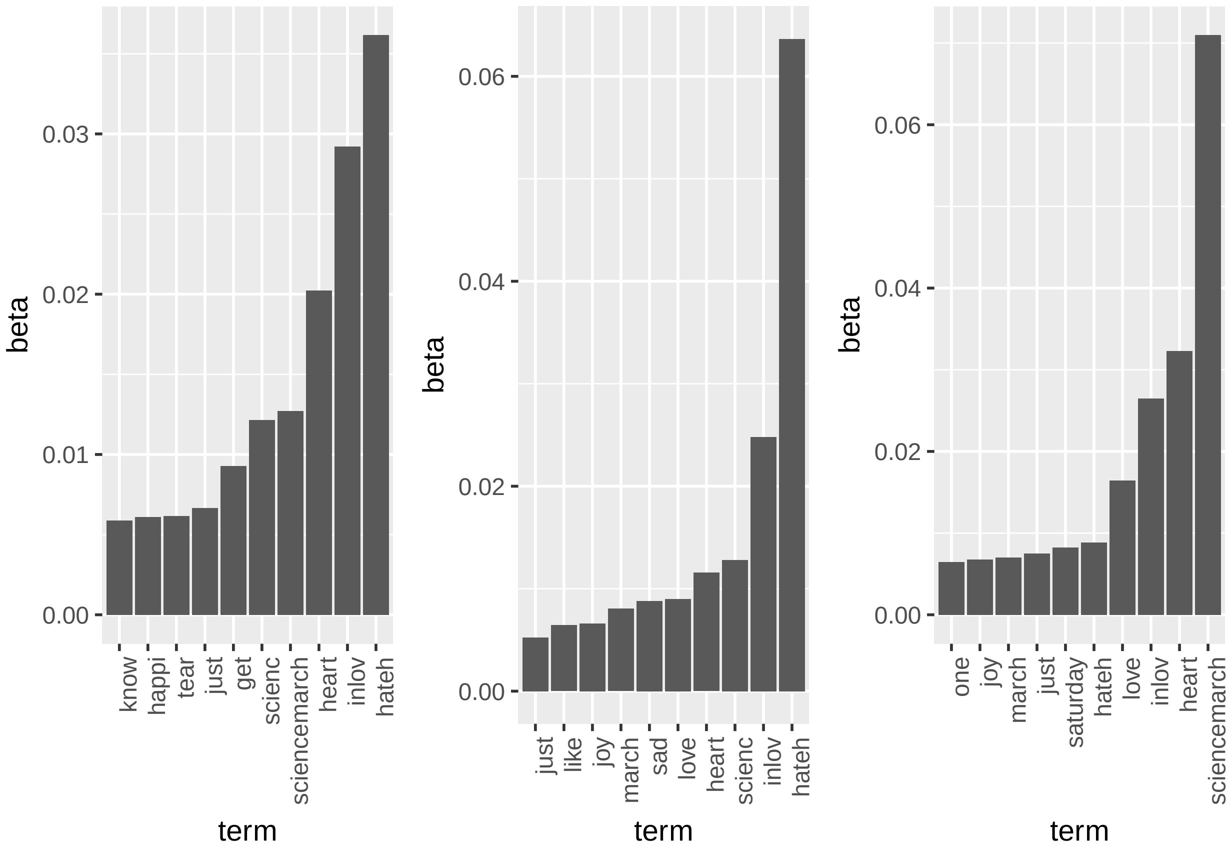
The last case was to perform LDA after translating the Unicode Emoji characters in English. `unicode` package was used to match the Unicode to its name. Then the standard process of stemming and deletion of stop words where performed.

Table 7: Output of LDA with translated Unicode

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

Table 8: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
hateh	0.03615	hateh	0.06366	sciencemarch	0.07096
inlov	0.02922	inlov	0.02479	heart	0.03231
heart	0.02023	scienc	0.0128	inlov	0.02648
sciencemarch	0.0127	heart	0.01156	love	0.01645
scienc	0.01216	love	0.009006	hateh	0.008874
get	0.009276	sad	0.008808	saturday	0.008227
just	0.006675	march	0.008049	just	0.007479
tear	0.006154	joy	0.0066	march	0.007012
happi	0.006088	like	0.006427	joy	0.006779
know	0.005881	just	0.005256	one	0.006454



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
uni_png: Image of emojis in PNG format represented in a matrix format

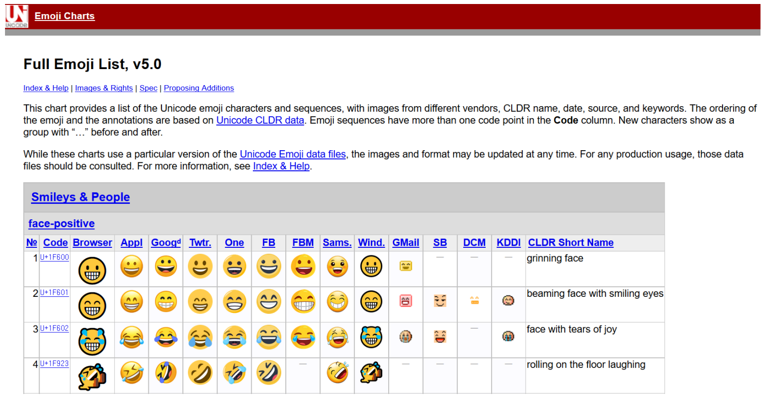


Figure 3: Glimpse of the table of Emoji on the Unicode.org website

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