Latent Dirichlet Allocation Models Considering Emojis

Taikgun Song

0.0.1 abstract

XXX write later XXX

Contents

1 Introduction

Latent Dirichlet Allocation(LDA) is a popular hierarchical Bayesian model that is widely used as a topic modeling method.

What is topic modeling? - go a bit more gently in your introduction.

LDA exploits statistical inference to discover latent topic of text data, however, the method depends on the number of observations - words.

You are sneaking the data in through the backdoor - slow down and describe the data first.

This dependency on observed words lead LDA to its systematic limit is when there exists data sparcity with short text data.

do you have a citation for that problem? Otherwise this is a statement that you would need to prove. That would be a distraction. Instead, you can argue that emojis are part of texts used on social media and are not used in the analysis.

New communication media such as Social Network Services (SNS) and User Generated Content (UGC) platform increase the amount of text data usage, however, the size of the document is limited to a couple hundred words. Hence, LDA model is known for its low performance on these short online text due to the data sparcity.

OK, I'm feeling very old. Give examples for SNSs - I also don't quite see why you are separating SNS from UGC. Isn't UGC by default what makes SNS?

Give some examples for tweets you scraped – that will automatically lead into the use of emojis. xxx Bridge xxx

The use of emojis - pictograms that express the author's feelings - mixed in with other text is a unique characteristic of online messages. Conventionally, Emoji characters have been considered as a noise and were deleted prior to applying LDA technique. slow down! In contrast with the previous procedure, this paper propose the idea of incorporating Emoji characters to enhance the performance of the LDA method on short online texts.

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.

XXX Should the packages used to run example be introduced here with brief steps? XXX The `tm`, `topicmodels`, `emoji`, `tidytext`, and `tidyverse` package in `R` was written to help the above analysis.

2 LDA

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:

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

where

Graphical display of LDA is given in Figure 1.

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.

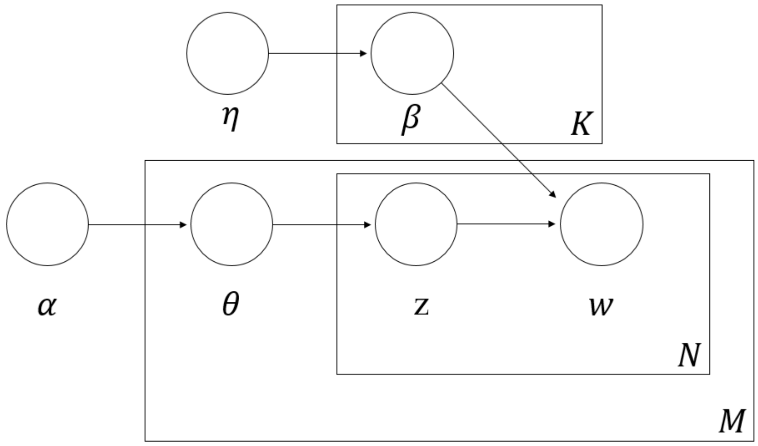


Figure 1: Graphical Model representation of LDA

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 processing LDA method, 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.

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

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

3 Application

3.1 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 illustraited in Table 1. 52.7% of the #inlove message strings, 29.3% of the #hateher message strings, and 7.8% of #marchscience message strings have one or more emoji information.

Table 1: Proportion of Twitter messages with Emoji

	#inlove	#hateher	#marchscience
Proportion	0.5275	0.2926	0.07782

For hashtag #inlove, total number of 1188 Emojis were used from 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 hastags

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 Application

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

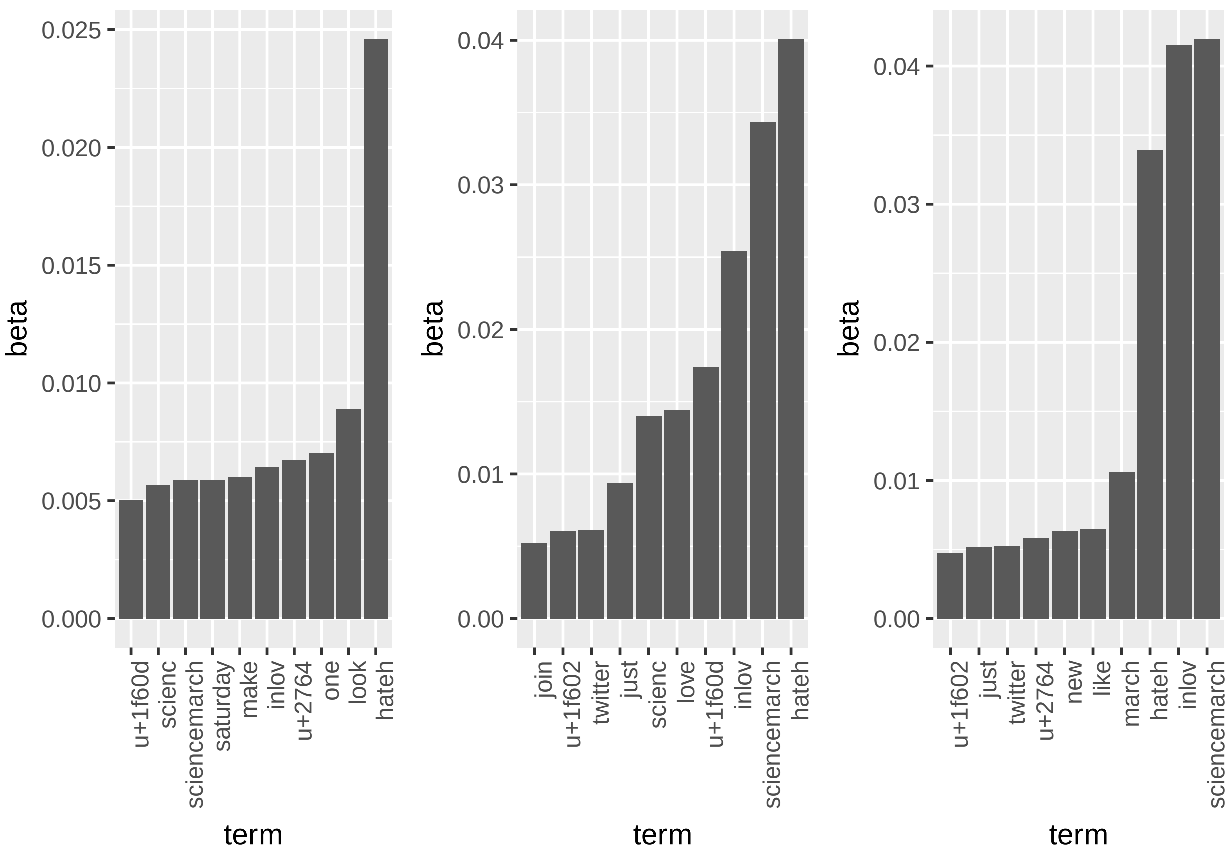
The second case was to run LDA on a raw data set. Stemming and stop word deletion were performed. Different number of topic dimensions were tested and the result of 4 topic dimension with 10 terms are provided in Table 3. Describe the output.

Table 3: Output LDA with the raw data

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

Table 4: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
hateh	0.0246	hateh	0.04008	sciencemarch	0.04196
look	0.008909	sciencemarch	0.03434	inlov	0.04151
one	0.007037	inlov	0.02543	hateh	0.03392
u+2764	0.006723	u+1f60d	0.01738	march	0.01061
inlov	0.006427	love	0.01443	like	0.006503
make	0.005992	scienc	0.01399	new	0.006327
saturday	0.005876	just	0.009383	u+2764	0.005834
sciencemarch	0.005865	twitter	0.006135	twitter	0.005274
scienc	0.005657	u+1f602	0.006029	just	0.005144
u+1f60d	0.005026	join	0.005253	u+1f602	0.004752



3.4 LDA without Unicode

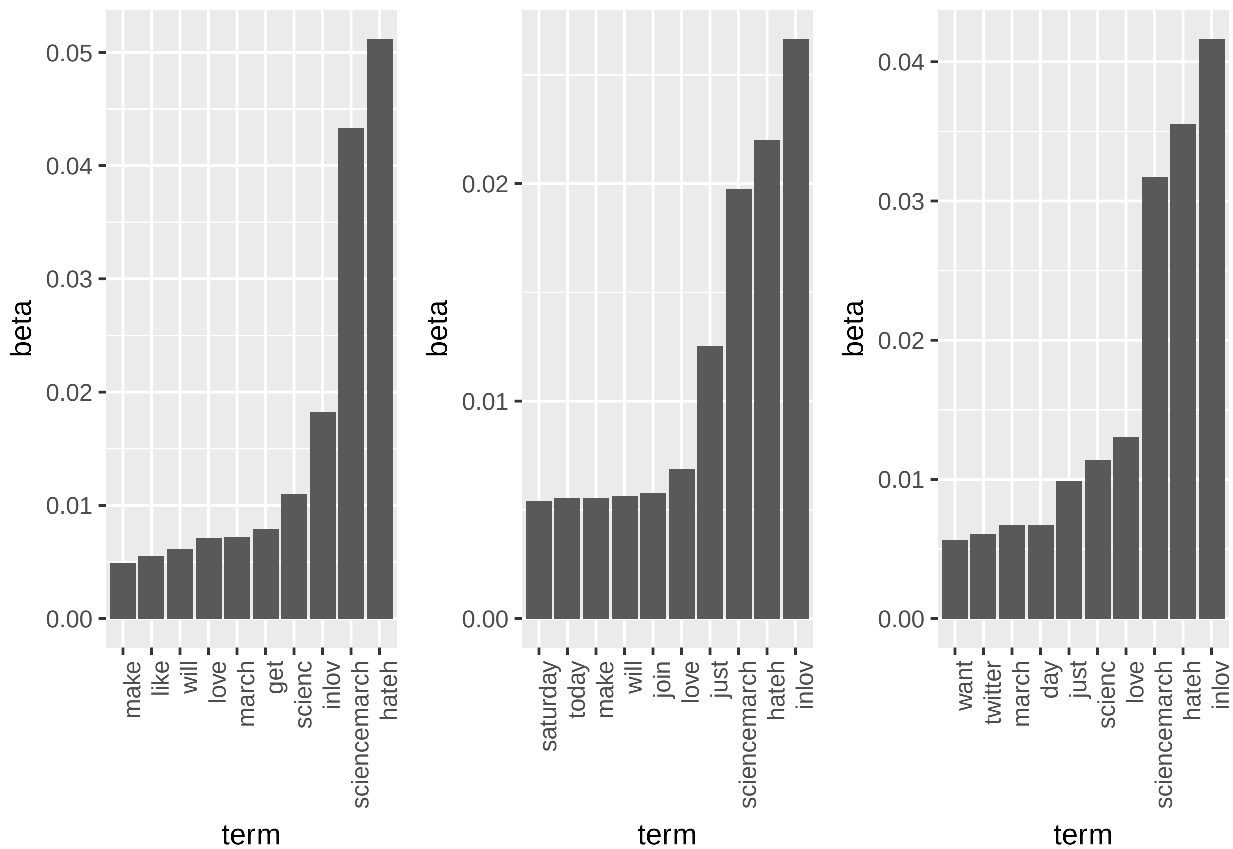
In most text mining examples, LDA is performed after removing the Unicode information. For the first case, therefore, Unicode characters were removed from the raw text data set. Then, the standard procedure of stemming and stop word deletion was performed to enhance the accuracy of LDA. `tm` package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

Table 6: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
hateh	0.05118	inlov	0.02665	inlov	0.04164
sciencemarch	0.04333	hateh	0.02201	hateh	0.03555
inlov	0.01827	sciencemarch	0.01977	sciencemarch	0.03173
scienc	0.01102	just	0.01253	love	0.01307
get	0.007933	love	0.006878	scienc	0.0114
march	0.007167	join	0.005778	just	0.00989
love	0.007092	will	0.005653	day	0.00672
will	0.006095	make	0.005561	march	0.006687
like	0.005539	today	0.005551	twitter	0.006047
make	0.004899	saturday	0.005413	want	0.005638



3.5 LDA with name translated

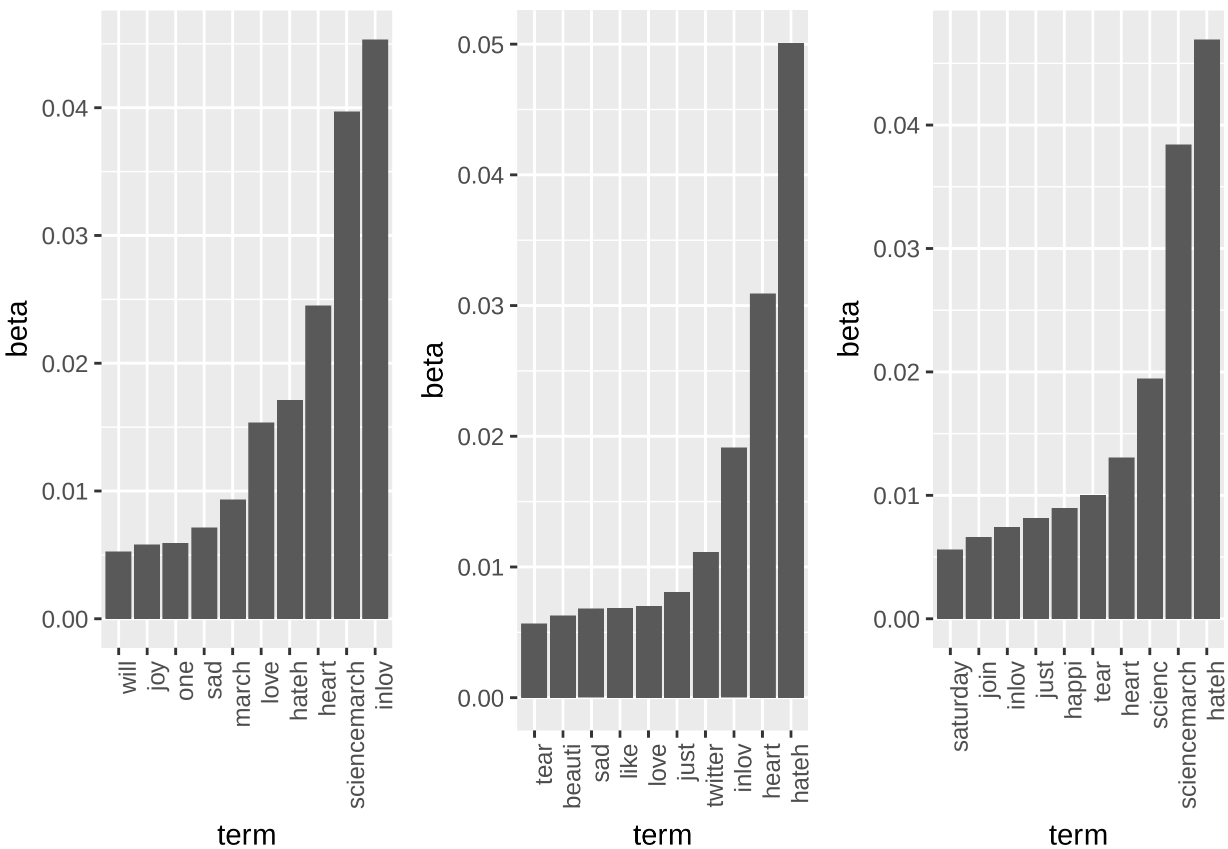
The last case was to perform LDA after translating the Unicode Emoji characters in English. `unicode` package was used to match the Unicode to its name. Then the standard process of stemming and deletion of stop words where performed.

Table 7: Output of LDA with translated Unicode

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

Table 8: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
inlov	0.04536	hateh	0.05008	hateh	0.04695
sciencemarch	0.03969	heart	0.03092	sciencemarch	0.03842
heart	0.0245	inlov	0.01912	scienc	0.01945
hateh	0.01714	twitter	0.01114	heart	0.01306
love	0.01537	just	0.008093	tear	0.01002
march	0.009318	love	0.007018	happi	0.008968
sad	0.007141	like	0.006873	just	0.008153
one	0.005942	sad	0.006809	inlov	0.007433
joy	0.005823	beauti	0.006295	join	0.006613
will	0.005257	tear	0.005683	saturday	0.005598



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 2.

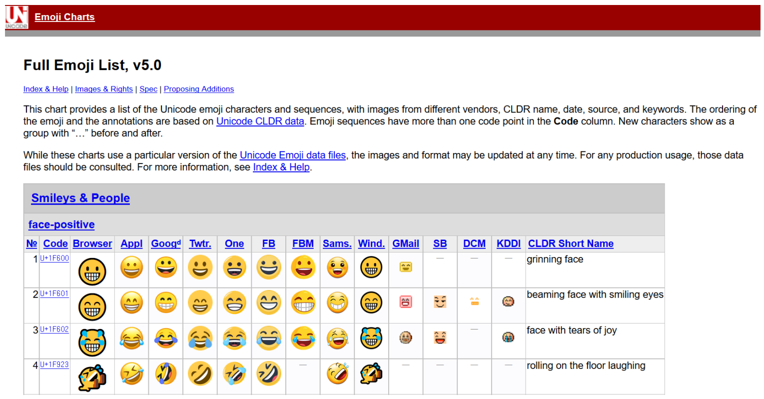


Figure 2: Glimpse of the table of Emoji on the Unicode.org website

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.

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