

Emoji

Tek

Introduction

The development of Social Network Service(SNS) and User Generated Content(UGC) platforms have been providing a new channels for its users to communicate and share opinions with the fellow community members. Hundreds and thousands of new messages are shared every day on Twitter and Facebook; product reviews are posted on Amazon and Trip Advisor. The text contents of SNS and UGC is a raw information of Individual's perception and emotion. As one would expect, this distinct characteristic of SNS and UGC text data could be a valuable information in different academic disciplines and industries. As a result, different machine learning and natural language processing techniques were developed to evaluate text data.

Most natural language processing(NLP) techniques require preparation procedures to improve its performance. Deleting stop words(determiners, conjunctions, prepositions, pronouns, auxiliary verbs, modals, and quantifiers) is an example of this preceding step. The result of this step increases the accuracy of the NLP by reducing the noise generated from words that does not convey any contextual meaning. In this context, Emoji(a pictographic information that carries class of feelings) in the text data has been considered as a noise and was deleted prior to applying NLP techniques.

Although deleting Emoji Unicode before NLP is a standard operation, unlike deleting stop words, this does take away information that might have modest contribution to the context. Emoji, originally driven from Japanese word e(picture) + moji(character), is a pictograph that has become widely used on internet web pages and on SNS plaforms. Emoji, much like its close relative emoticon, could provide visual representation of not only solid objects, but also emotions through facial expressions and symbols related to feelings and moods. Communication via traditional text characters, such as words in alphabets, may not be as effective and efficient as Emojis when conveying emotions. For this reason, Emoji gained popularity after 1990 especially after cell phones and internet came to wide use. Therefore, simply deleting Emoji would reduce the information contained in the original text data.

In contrast to filtering Emoji characters out, reflecting Emoji information during the evaluation have benefits. First, eacho Emoji has pre-determined topic dimension set by the official organization. Therefore, crude topic matching may be accomplished using the Emoji information. Second, Emoji may be helpful for sentiment analysis. Being closely related to emotions and mood, sentiment analysis on Emoji will provide auxiliary information.

The `emoji` package in R was written to help the above analysis.

Emoji Data Set

`emoji` package in R

Plan to change this part after posting the `Emoji` package on CRAN

Emoji in a text data is encoded as a sequence of Unicode: an industrial standard that consists encoding, representation, and text expression of writing system. *The Unicode Standard* is distributed by a non-profit organization the *Unicode Consortium*. The current list of Emoji v5.0 is available on the official *Unicode Consortium* website. Example illustration of the Emoji table is attached in Figure 1. Data set of Emoji characters are available in `emoji` package.








































Smileys & People															
face-positive															
Nr	Code	Browser	App	Goog	Twtr	One	FB	FBM	Sams	Wind	GMail	SB	DCM	KDDI	CLDR Short Name
1	U+1F600														grinning face
2	U+1F601														grinning face with smiling eyes
3	U+1F602														face with tears of joy

Figure 1: Glimpse of the table of Emoji on the unicode.org website

Generate different type of encodings using Python

A script was written R that changes between different encoding environment. Plan to include this feature in the '`emoji`' package. There are multiple way of encoding Emojis on website or SNS. Unicode, Unicode escape, UTF-8hex,

zerox notation, NCR are examples of commonly used encodings. As shown in Figure 1, the initial data set scraped from the *Unicode Consortium* only has Unicode. The most common encoding type for online web page, however, is UTF-8 and Unicode escape. Therefore, the original Unicode sequence should be translated into different encoding types for the data set to be applied. Different types of encoding format were generated from the original Unicode via simple Python code.

Scoring

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.

Description of the Emoji Data set

The table should be updated. It is using the v4.0 Emoji list

The complete Emoji data set is saved under the ‘data’ directory. This complete data set is read and named ‘uni_info’. Some Emojis are a combination of two or more basic Emojis. For example, Emoji ‘boy: light skin tone’ is a combination of ‘boy’ (U+1F466) and ‘light skin tone’ (U+1F3FB). Data ‘basic_uni_info’ is a data set of the basic Emojis. There are many different ways to encode ‘Unicode’. The data set includes the following encoding types: ‘U+hexadecimal’, ‘UTF-8 hexadecimal’, ‘hexadecimal’, and ‘numeric character reference (NCR)’. The example of the data set is given in Table 1

Table 1: Information of 5 Emoji data set

uni_No	uni_code	uni_name	uni_age	uni_keyws	utf_8_hex	zerox_notation	ncr	PosScore	NegScore
1	U+1F600	grinning face	2012	face	f09f9880	0x1f600	128512	0	0
1	U+1F600	grinning face	2012	face	f09f9880	0x1f600	128512	0	0.5
1	U+1F600	grinning face	2012	face	f09f9880	0x1f600	128512	0.125	0.125
1	U+1F600	grinning face	2012	face	f09f9880	0x1f600	128512	0.125	0.375
1	U+1F600	grinning face	2012	grin	f09f9880	0x1f600	128512	0	0

Application

Data Set and exploratory data analysis

Two samples of twitter messages with the following hastag #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 2. 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 2: 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 3.
















#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 3: 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.

LDA

To address the importance or reflecting the emoji information in text data, Latent Dirichlet Allocation (LDA): a popular topic modeling method, was performed on twitter messages scraped online. 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.

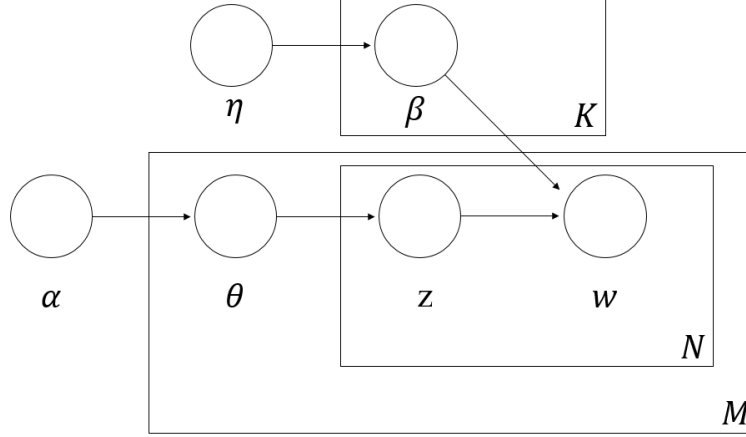


Figure 2: Graphical Model representation of LDA

LDA Equation goes here

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

LDA on a raw data set

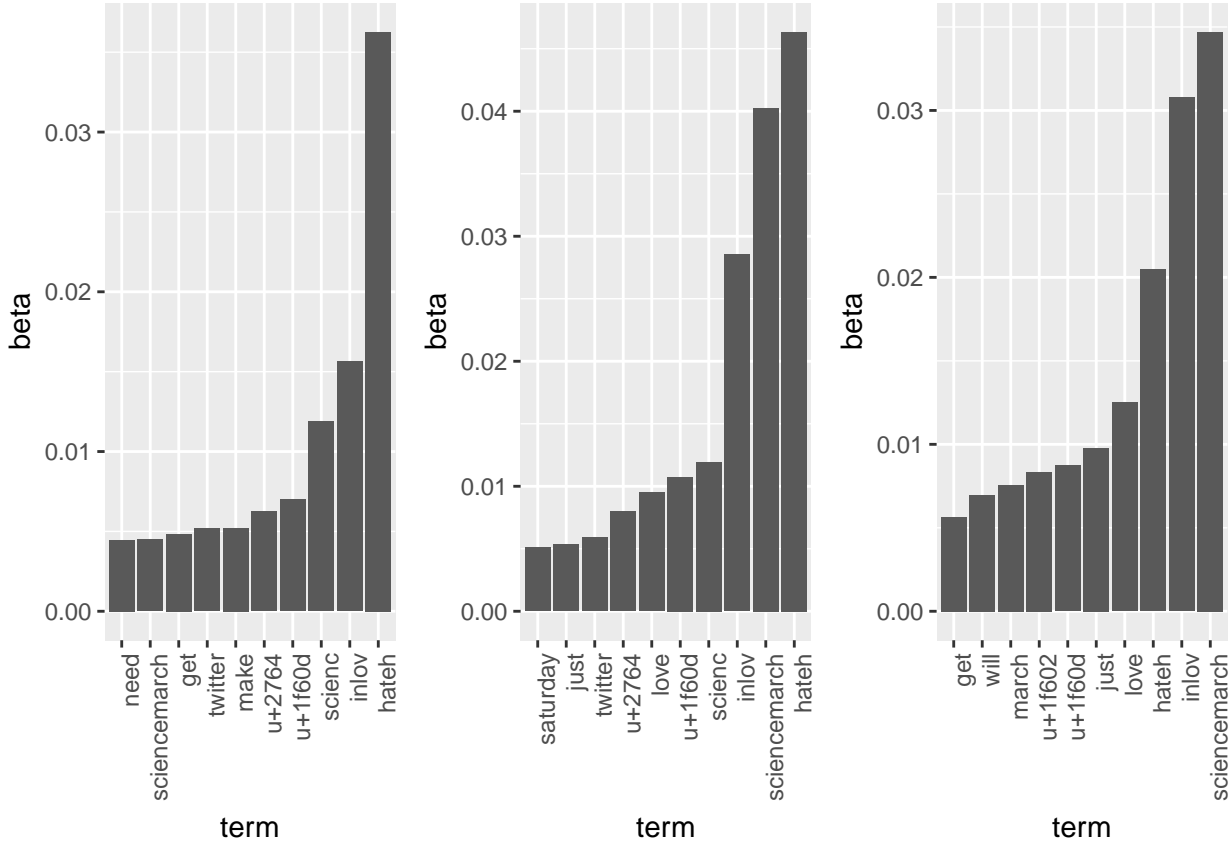
The second case was to run LDA on a raw data set. Stemming and stop word deletion were performed. Different number of topic dimensions were tested and the result of 4 topic dimension with 10 terms are provided in Table 4. Describe the output.

Table 4: Output LDA with the raw data

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

Table 5: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
hateh	0.03628	hateh	0.04636	sciencemarch	0.03471
inlov	0.01564	sciencemarch	0.04027	inlov	0.03077
scienc	0.01187	inlov	0.02854	hateh	0.02048
u+1f60d	0.007005	scienc	0.01193	love	0.0125
u+2764	0.006245	u+1f60d	0.01075	just	0.009775
make	0.005205	love	0.009508	u+1f60d	0.008723
twitter	0.005187	u+2764	0.007966	u+1f602	0.008357
get	0.004822	twitter	0.005899	march	0.007565
sciencemarch	0.004478	just	0.005355	will	0.006959
need	0.004466	saturday	0.005092	get	0.005652



LDA without Unicode

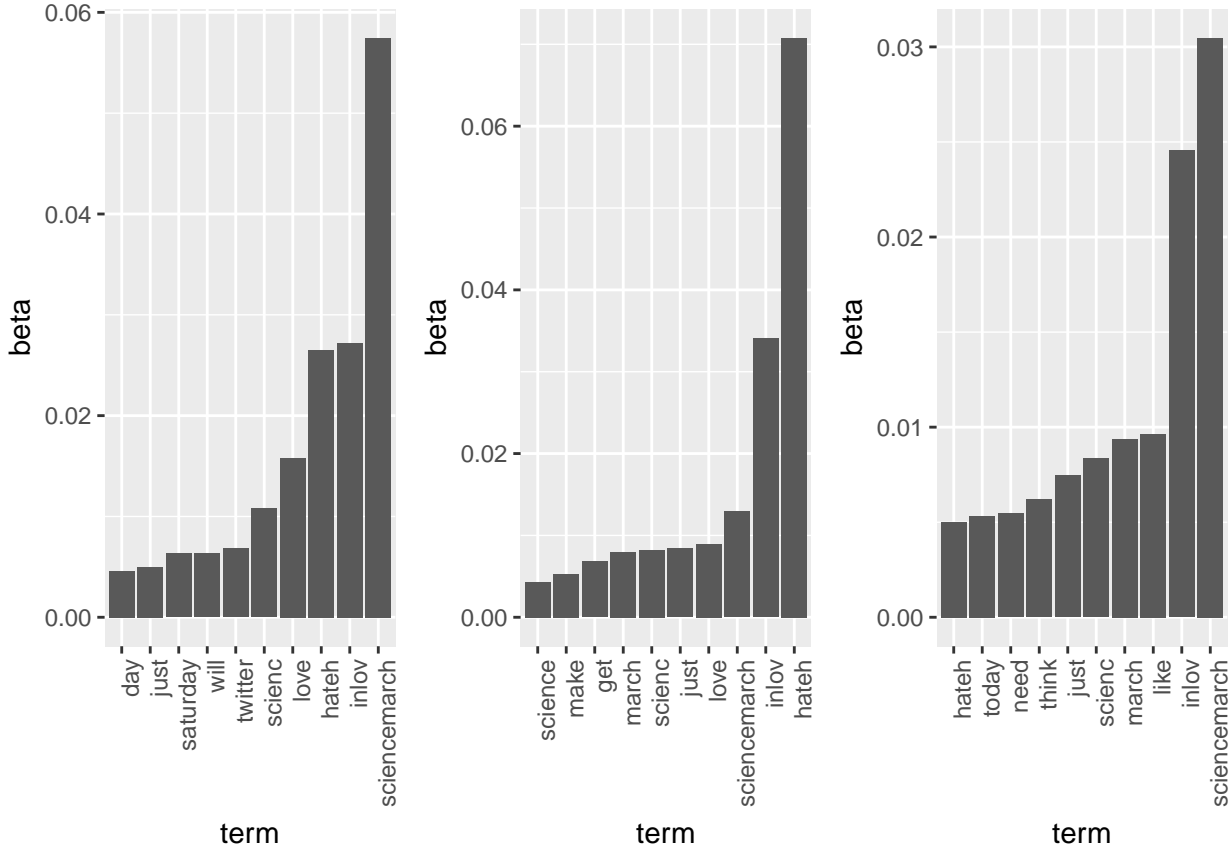
In most text mining examples, LDA is performed after removing the unicode information. For the first case, therefore, unicode characters were removed from the raw text data set. Then, the standard procedure of stemming and stop word deletion was performed to enhance the accuracy of LDA. `tm` package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

Table 6: Output of LDA with the raw data without the Unicode

Topic 1	Topic 2	Topic 3
sciencemarch	hateh	sciencemarch
inlov	inlov	inlov
hateh	sciencemarch	like
love	love	march
scienc	just	scienc

Table 7: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
sciencemarch	0.05748	hateh	0.0708	sciencemarch	0.03048
inlov	0.02718	inlov	0.03405	inlov	0.02455
hateh	0.02649	sciencemarch	0.01293	like	0.009637
love	0.0158	love	0.008934	march	0.009375
scienc	0.01077	just	0.00842	scienc	0.008347
twitter	0.006833	scienc	0.008211	just	0.007455
will	0.006342	march	0.008	think	0.006182
saturday	0.006319	get	0.006832	need	0.005457
just	0.004984	make	0.005271	today	0.005325
day	0.004604	science	0.004327	hateh	0.005013



LDA with name translated

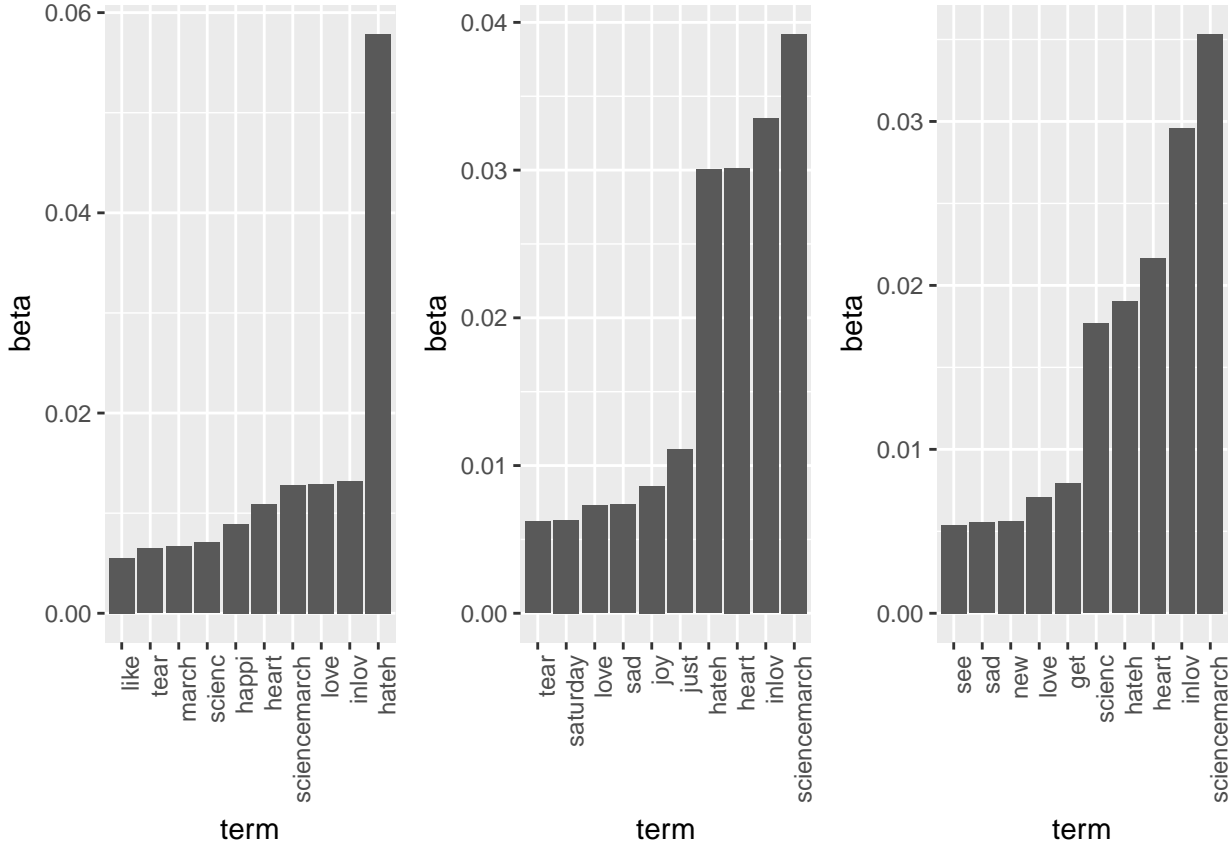
The last case was to perform LDA after traslating the unicode Emoji characters in english. `unicode` package was used to match the unicode to its name. Then the standard process of stemming and deletion of stop words where performed.

Table 8: Output of LDA with translated Unicode

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

Table 9: Word prob. given topic

1.term	1.beta	2.term	2.beta	3.term	3.beta
hateh	0.05789	sciencemarch	0.03923	sciencemarch	0.03535
inlov	0.01321	inlov	0.03349	inlov	0.02959
love	0.01291	heart	0.03013	heart	0.02167
sciencemarch	0.01281	hateh	0.03003	hateh	0.01901
heart	0.01093	just	0.01108	scienc	0.01768
happi	0.008898	joy	0.008608	get	0.007942
scienc	0.007082	sad	0.007361	love	0.007106
march	0.006649	love	0.007288	new	0.00563
tear	0.006474	saturday	0.006321	sad	0.005532
like	0.005532	tear	0.006209	see	0.005375



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.