Course Reminders

Due tonight (Wed): A3

Due Sun: D6, Q7, Checkpoint #1, Weekly Project Survey (*optional*)

Notes: Project Proposal Feedback Now Available

- Scores and Feedback on GitHub repo (as an issue)
  - Rubric used (see issue)
  - Score on canvas should be the same as on GH (but if your score is a zero on Canvas, suggests you did not participate in proposal completion - let's chat if you disagree)
- No official "regrades" to submit"
  - If you agree with feedback, make changes/improvement to Data Checkpoint Notebook
  - If you disagree with feedback, leave a comment on the project proposal GH Issue
  - (Leave Proposal notebook as is)
  - Scores will automatically update for initial points lost upon data checkpoint grading
- Pivoting/changing course is typical/part of the process!

Text Analysis

Shannon E. Ellis, Ph.D
UC San Diego

Department of Cognitive Science
sellis@ucsd.edu

Examples of questions that require text analysis

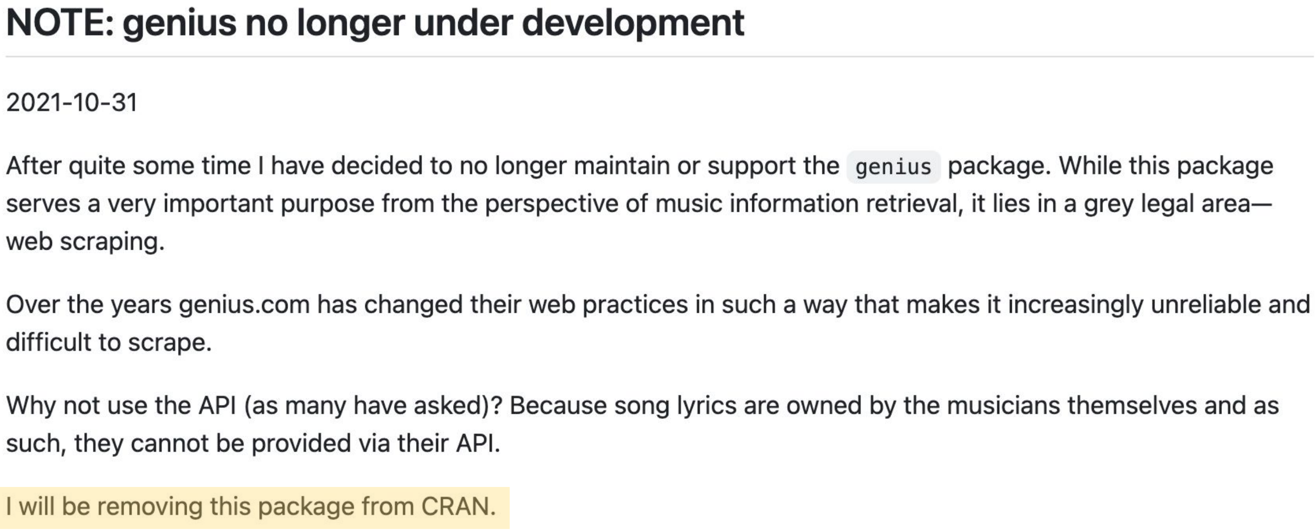
1. Did J.K. Rowling write The Cuckoo’s Calling under the pen name Robert Galbraith?
2. What themes are common in 19th century literature?
3. Can we tell the difference between tweets that come from Trump himself or a staffer?
4. Is Hillary the most poisoned name in US History?
5. Can we visualize the narrative structure of The Hobbit?
6. Who has the biggest vocabulary in hip hop?
7. Is there a gender imbalance in who gets to say lines in the movies?

Text Ideas/Processing in Projects

- Spam detection (in text messages, on social media)
- Hate speech detection
- Predictive text (like gmail)
- Summarize articles (TL;DR)
- Summarize trends on social media… or in cutting edge NLP research?
- Plagiarism detection… in code projects?
- Document similarity… prevent the posting of duplicate questions on Campuswire?

Conceptual example...

I *used* to walk through text analysis of song lyrics over time, analyzing the sentiment and uniqueness of words, including the most popular current songs, which was honestly the only way I knew what songs/artists were popular but...



NOTE: genius no longer under development

2021-10-31

After quite some time I have decided to no longer maintain or support the `genius` package. While this package serves a very important purpose from the perspective of music information retrieval, it lies in a grey legal area—web scraping.

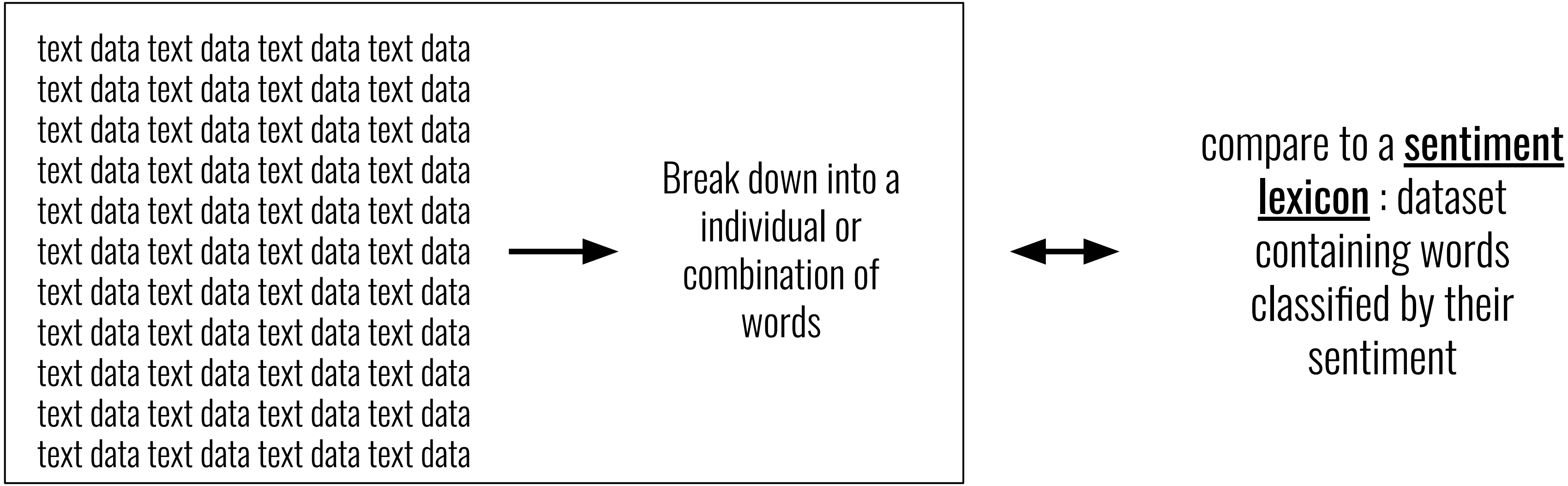
Over the years genius.com has changed their web practices in such a way that makes it increasingly unreliable and difficult to scrape.

Why not use the API (as many have asked)? Because song lyrics are owned by the musicians themselves and as such, they cannot be provided via their API.

I will be removing this package from CRAN.

Sentiment Analysis

Programmatically infer emotional content of text

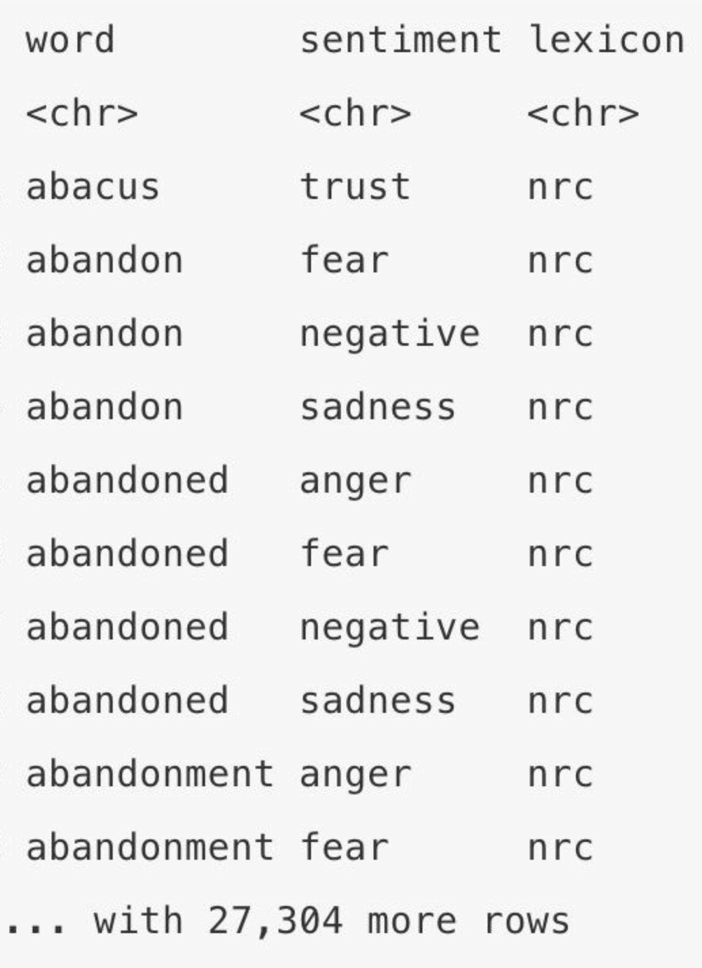


text data text data text data text data
text data text data text data text data
text data text data text data text data
text data text data text data text data
text data text data text data text data
text data text data text data text data
text data text data text data text data
text data text data text data text data
text data text data text data text data
text data text data text data text data
text data text data text data text data

Break down into a individual or combination of words

compare to a **<u>sentiment lexicon</u>** : dataset containing words classified by their sentiment

Example of sentiment **lexicon**:



```
word        sentiment lexicon
<chr>       <chr>     <chr>
abacus      trust     nrc
abandon     fear      nrc
abandon     negative  nrc
abandon     sadness   nrc
abandoned   anger     nrc
abandoned   fear      nrc
abandoned   negative  nrc
abandoned   sadness   nrc
abandonment anger     nrc
abandonment fear      nrc
... with 27,304 more rows
```

http://varianceexplained.org/r/trump-tweets/

When doing sentiment analysis...

token - a meaningful unit of text

- what you use for analysis
- *tokenization* takes corpus of text and splits it into tokens (words, bigrams, etc.)

stop words - words not helpful for analysis

- extremely common words such as “the”, “of”, “to”
- are typically removed from analysis

https://www.tidytextmining.com

When doing sentiment analysis...

stemming or **lemmatization**

Identifying the root for each token

Jumping, jumped, jumps, jump all have the same root ‘jump’

Where things get tricky: jumper???

Stemmers and lemmatizers do the same thing, stemmers are word focussed/cruder and lemmatizers try to take into account the context to correctly stem jumper differently depending on if its “I love my new jumper” (In the US jumper=a kind of dress, in the UK jumper=sweater) or its “Shane is a long jumper”

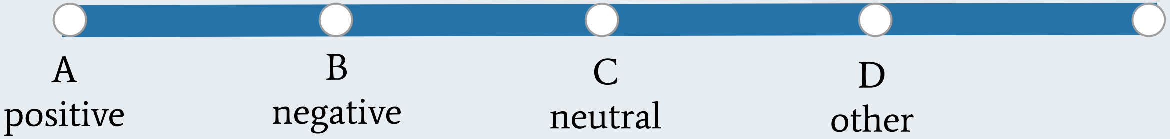
In text analysis, your choices matter:

1. How to tokenize?
2. What lexicon to use?
3. Remove stop words? Remove common words?
4. Use stemming?

Sentiment Limitations

How would you classify the sentiment of the following sentence?

“The idea behind the movie was great, but it could have been better”



A
positive

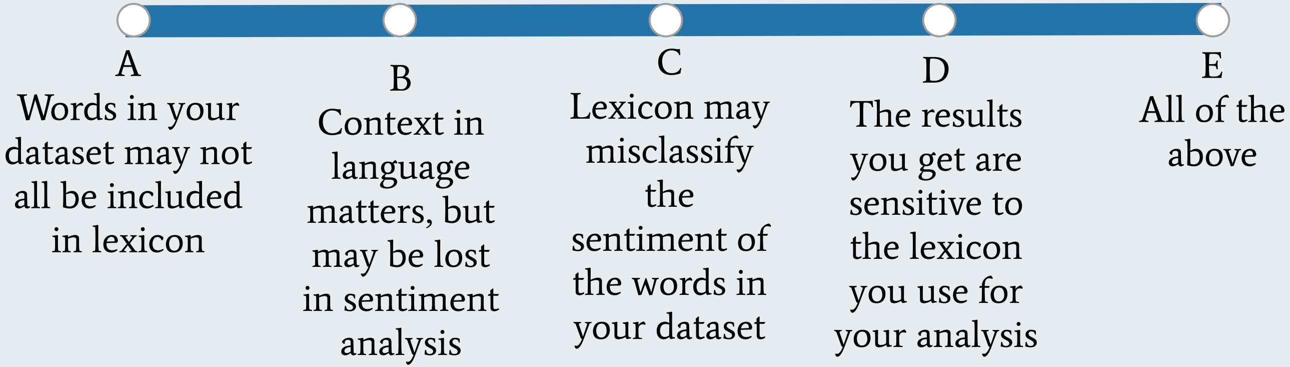
B
negative

C
neutral

D
other

Sentiment Limitations

What is a limitation of sentiment analysis?



A
Words in your dataset may not all be included in lexicon

B
Context in language matters, but may be lost in sentiment analysis

C
Lexicon may misclassify the sentiment of the words in your dataset

D
The results you get are sensitive to the lexicon you use for your analysis

E
All of the above

TF-IDF:

Term Frequency - Inverse Document Frequency

Term Frequency (TF) : how frequently a word occurs in a document

Inverse document frequency (IDF) : intended to measure how important a word is to a document

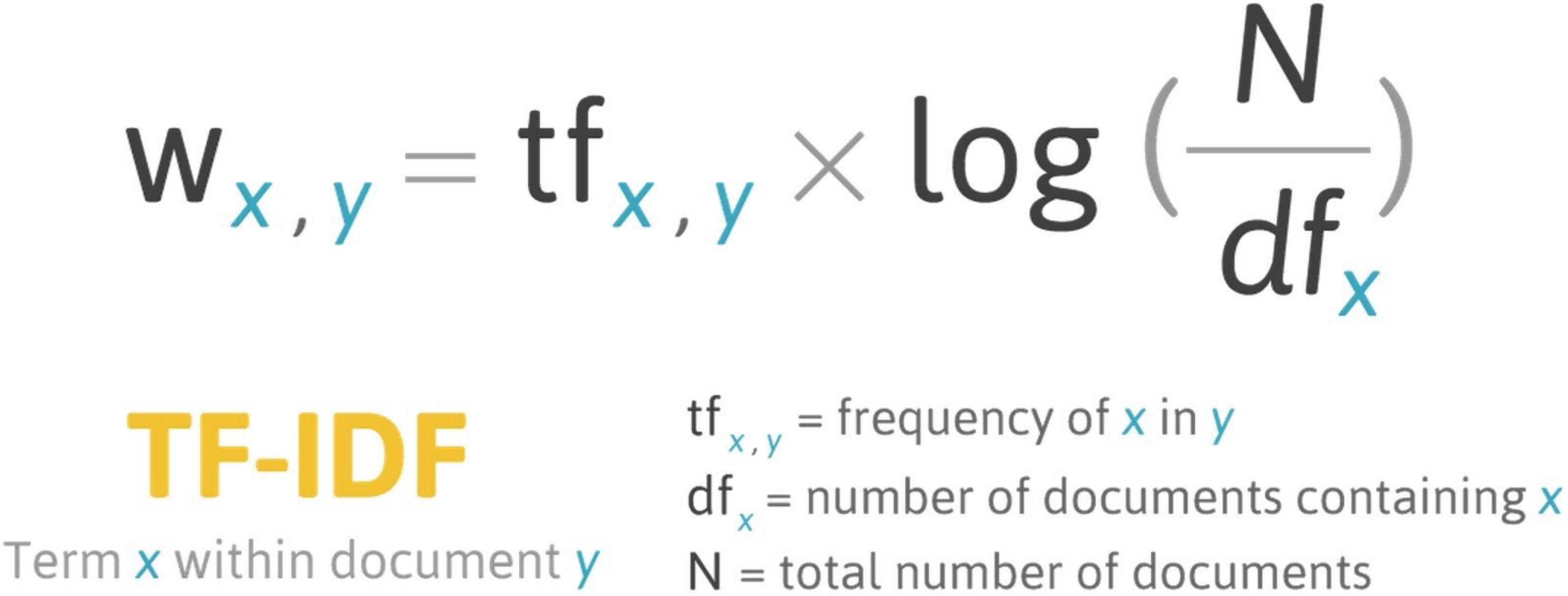
decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents

$$idf(\text{term}) = \ln\left(\frac{n_{\text{documents}}}{n_{\text{documents containing term}}}\right)$$

TF-IDF:

Term Frequency - Inverse Document Frequency

the frequency of a term adjusted for how rarely it is used



$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

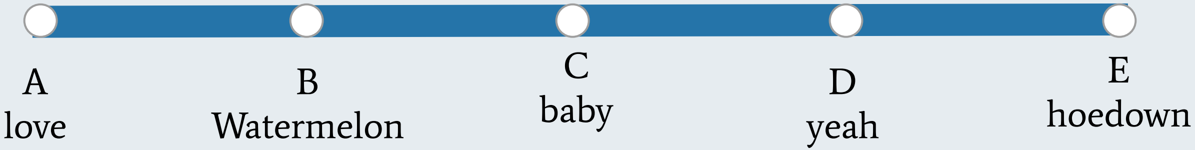
df_x = number of documents containing x

N = total number of documents

https://www.tidytextmining.com

TF-IDF

What would you guess may be the word with a high TF-IDF value in a dataset looking at lyrics across years ?



A
love

B
Watermelon

C
baby

D
yeah

E
hoedown

Text Analysis: The Big Picture

- Identify the problem you are trying to solve... what you want to answer suggests the approaches you will use below
- Clean and preprocess the data
- Linguistic transformations
  - Tokenization, lemmatization, stemming, remove stop words
  - [optional] part of speech tagging, named entity recognition
- EDA
- [optional] Calculate measurements and statistics
  - e.g., diversity, density, sentiment, TF-IDF
- [optional] Transform representation
  - Bag of words, vector embeddings
- [optional] Train a ML system or write an algorithm to accomplish a task given the representation
  - e.g., detect spam