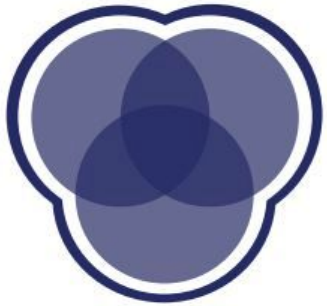


CDS

Cornell Data Science

Text Mining

Unstructured Data

So far, we've seen **structured** data. But how do you analyze this?

```
Serv. God gi' go-den. I pray, sir, can you read?

 Rom. Ay, mine own fortune in my misery.

 Serv. Perhaps you have learned it without book. But I pray, can
   you read anything you see?

 Rom. Ay, If I know the letters and the language.

 Serv. Ye say honestly. Rest you merry!

 Rom. Stay, fellow; I can read.                        He reads.

     'Signior Martino and his wife and daughters;
     County Anselmo and his beauteous sisters;
     The lady widow of Vitruvio;
     Signior Placentio and His lovely nieces;
     Mercutio and his brother Valentine;
     Mine uncle Capulet, his wife, and daughters;
```

Shakespeare, William. "Romeo and Juliet."

Document Model

We'd like to convert *words* into *features* so that we can use our existing models. But we can't lose too much information!

Bag of words model:

- **Document**: unordered multiset of important words
- **Corpus**: set of documents

https://ae01.alicdn.com/kf/HTB1ifrjJFXXXXbEXFXXq6xXFXXXs/100pcs-bag-5-5cm-font-b-marine-b-font-font-b-ball-b-font-colored-children.jpg

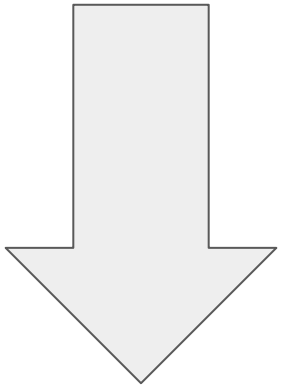
Question:

When would we not want to use bag-of-words?

Pros	Cons
• Simple to implement • Performs well on quick comparisons • Efficient	• Loses meaning of original document • Ordering of words doesn't matter (we lose idioms / pointwise mutual information)

Example

“Cornell Data Science is a data science project team.”



```
{“cornell”, “data”, “data”,
“project”, “science”, “team”}
```

How do we get there?

- Remove **stop words** like “the”, “it”, “and”, …
- Remove **punctuation** and **capitalization** (“cORNELL!” and “Cornell.” should be equivalent)
- [More advanced] Use **stemming** to reduce words to their roots

Cornell Data Science is a project-team-oriented organization that seeks to help students gain hands-on experience with data analytics and machine learning.	cornel data scienc is a project team orient organ that seek to help student gain hand on experi with data analyt and machin learn

http://9ol.es/porter_js_demo.html

Document-Term Matrix

Assume we have several **documents** that we convert to bags of words. Can organize into matrix:

- Rows are documents
- Columns are processed words (**terms**)

Cell value *n* signifies that a term appears in the document *n* times.

	cornell	**data**	**science**
D_1	1	0	1
D_2	2	1	0

Predictive Coding

Documents are data points; frequencies of terms are features. We want to predict which documents are **responsive** to a given **query**.

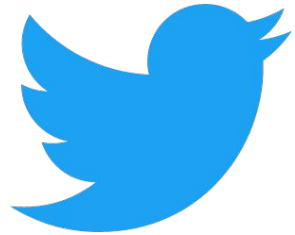
This is just binary classification on documents!

CART + Bag of Words

We will use a corpus of tweets from election night.

We've created a data set that contains two sets of tweets:

- Training set: manually labeled as pro-Trump
- Test set: use decision trees to determine whether each tweet is pro- or anti-Trump



Predictive Coding Demo

Demo time!

Sentiment Analysis

We can also see how emotionally charged a piece of text is.

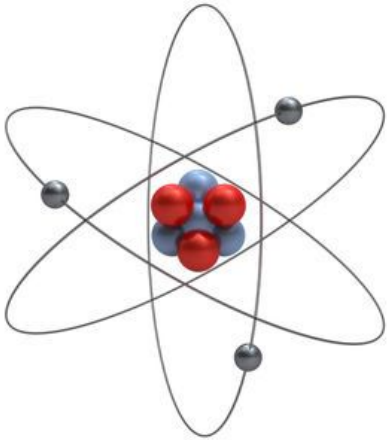
Common Approach: Word Sentiment

Each word is associated with a specific "charge" (positive or negative).

Document sentiment is the sum of word sentiments.

"Great" = +1

"Horrible" = -1



Question:

Name some disadvantages of this approach.

Sentiment Analysis Demo

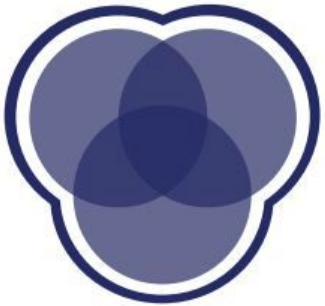
We'll be using the `tidytext` package to perform analysis of word sentiment across several documents.

Coming Up

Your problem set: Continue final project

Next week: Our data's about to get bigger.

See you then!



CDS

Cornell Data Science