In [20]:
```
from notebook.services.config import ConfigManager
cm = ConfigManager()
cm.update('livereveal', {
        "width": 1024,
        "height": 600,
        'scroll': True,
})
```

Out[20]: {'width': 1024, 'height': 600, 'scroll': True}

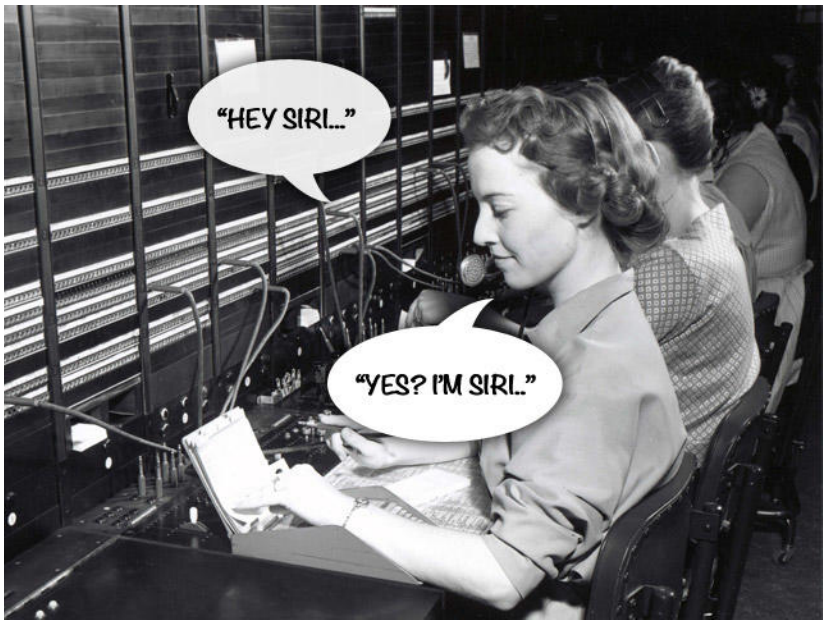
Text as Data

Fundamentals

Snorre Ralund

Natural Language Processing: State-of-the-Art (1)

Speech to Text



Human Reasoning

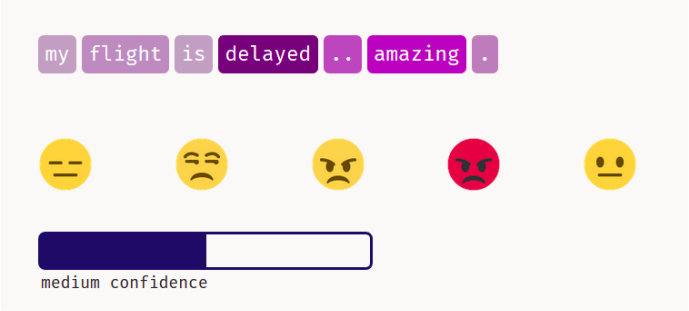
Sentiment Analysis

Deep Transfer Learning

Example: DeepMoji Felbo et al. 2017

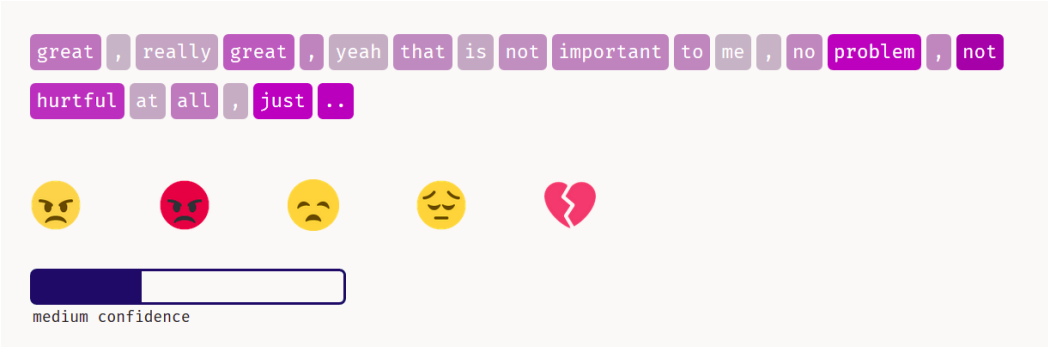
In [24]:
```
from IPython.display import Image
Image(filename = '/home/snorre/Dropbox/Forskning/PhD/undervisning
/images_lecture11/deepmoji_sarcasm.png')
```

Out[24]:



In [28]:
```
Image(filename = '/home/snorre/Dropbox/Forskning/PhD/undervisning
/images_lecture11/deepmoji_sarcasm2.png')
```

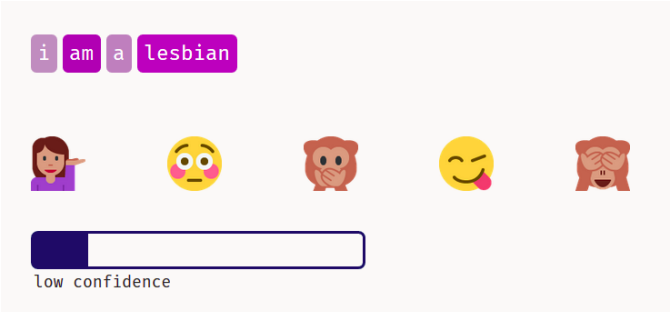
Out[28]:



Potential Bias

In [27]:
```
Image(filename='/home/snorre/Dropbox/Forskning/PhD/undervisning/i
mages_lecture11/deepmoji_bias.png')
```

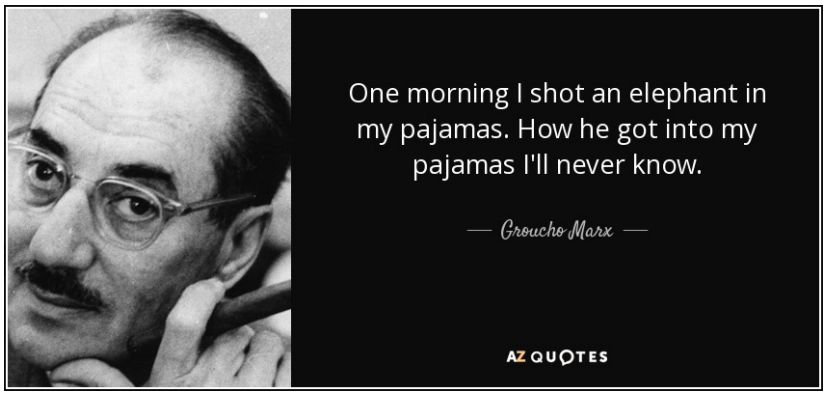
Out[27]:



Parsing of Natural Language

grammar and relation

- Who is present - Named-entity-recognition
- Who does what and to whom.
  - Pos tags.
  - noun - subject
  - verb - object
- Evaluations and Modifiers.
  - Adjectives
  - Adverbs.

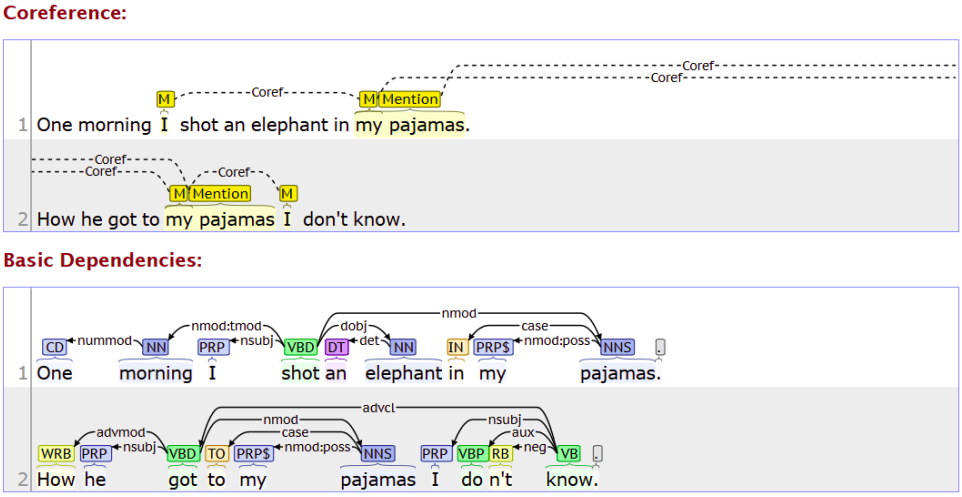


Natural Language Processing Frameworks

StanfordNLP

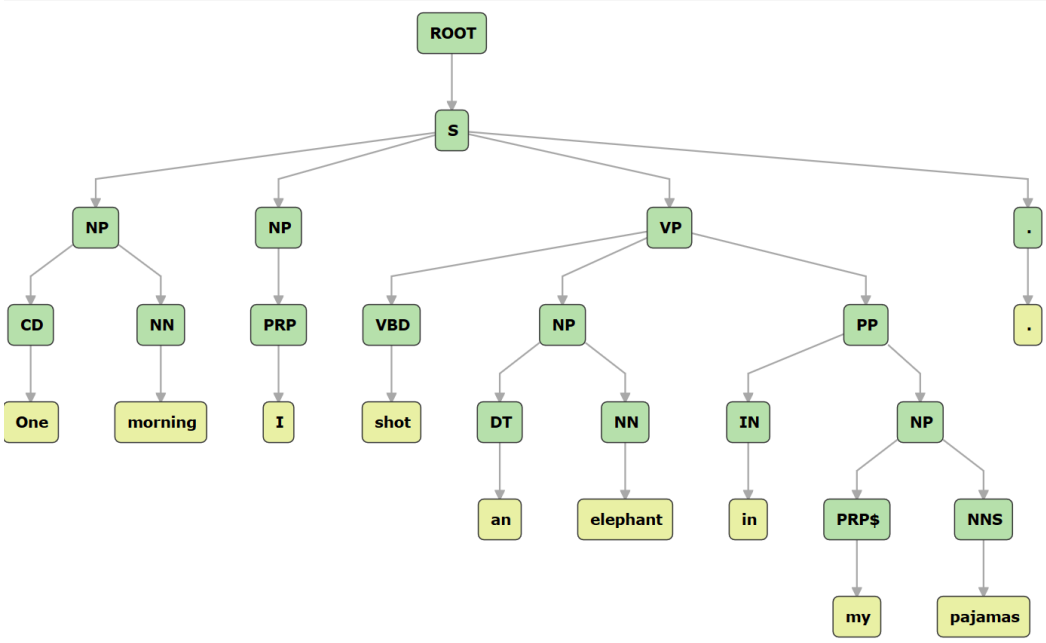
In [9]:
```
from IPython.display import Image
Image(width=800, filename='/home/snorre/Dropbox/Forskning/PhD/und
ervisning/images_lecture11/stanford_pajamas.png')
```

Out[9]:



In [10]:
```
Image(width=800,filename='/home/snorre/Dropbox/Forskning/PhD/unde
rvisning/images_lecture11/stanford_pajamas_constituency_parsing.p
ng')
```

Out[10]:



NLP in the Social Science

What do Social Scientist Do? (1)

Manual content analysis

Manual reading, interpretation and systematic annotation of texts.

Lasswell 1941: "The World Attention Survey"

Krippendorf 2018: Content Analysis: An Introduction to Its Methodology

- Let humans do what they do best.

BUT,

- High costs cannot scale to big data
- Problems with training coders and in turn intercoder reliability.

What do Social Scientist Do? (2)

Dictionary methods and Word Counts

Sentiment Lexicons (LIWC), Topic Lexicions, Argument Lexicons etc.

Examples

Especially in Psychology:

- Pennebaker and King 1999: "Linguistic styles: Language use as an individual difference."
- Graham, Haidt Nosek 2009: "Liberals and conservatives rely on different sets of moral foundations"
- Identification of suicidal behavior
- Menstrual Cycles inferred from autocorrelation in Emotional tweets.

Very efficient, and simple to set up

But,

- are they just random conglomerates with no consistent precision nor recall?
- Word ambivalences, and polysemi.
- Theoretical Validity? Is e.g. Negative emotions (i.e. sad and angry) actually a meaningful Category?

What do Social Scientist Do? (3)

Topic Modelling as measurement devices

- Cheap and scalable(...)
- Datadriven and inductive categories
- Capture Polysemi (...).

BUT

- Arbitrary models and hyperparameter choices.
- Also needs explicit validation - i.e. Test Set - to prove its validity

Examples

- Grimmer 2012: *"Appropriators not position takers"*: The distorting effects of electoral incentives on congressional representation"
- Fligstein et al. 2017: *"The hegemony of a macro-economic framework in the Federal Open Market Committee during and before the financial crisis of 2008"*

What should social Data Scientist Do? (1)

State-of-the-art Deep Learning.

- Learning from Bags-of-Words and atomized representations will not allow one to create high performing classifiers.

INSTEAD

- Use transfer learning (OpenAI, DeepMoji, ELMO, BERT).

Examples

- "Unsupervised Sentiment Neuron":https://openai.com/blog/unsupervised-sentiment-neuron/ (https://openai.com/blog/unsupervised-sentiment-neuron/)
- "DeepMoji": https://deepmoji.mit.edu/ (https://deepmoji.mit.edu/)

What should social Data Scientist Do? (2)

Computationally grounded analysis: Datadriven grounding of categories.

Social scientists cannot rely on categories / "theoretical schemata" developed ad hoc in other sciences. cf. Sentiment analysis.

INSTEAD

- We need a datadriven empirical understanding of what is important in the data.
- Using computational models (e.g. topic modelling) to effectively learn from large text data, and creating valid and empirically grounded categories.

What should social Data Scientist Do? (3)

Unbiased measurement and artificial effects

When using supervised learning as a Measurement device beating state-of-the-art in classifier performance is not important to the social scientist.

INSTEAD

- We need to know and estimate the bias of our model, and correct it (see e.g. Hopkins & King 2010).
- We need to make sure that our models do not systematically underestimate certain categories (e.g. social class, gender, ethnicity etc).

What should social Data Scientist Do? (4)

Ethics and Conscience

When developing more and more capable and sophisticated models to beat the state-of-the-art, one can lose sight of the consequences of the technology.

Instead Social data scientists appropriating these methods should:

- Be aware of the implications of your work
  - Who benefit from it, how can it be used and misused.
- Survelance and privacy

Examples:

- While Humans will not keep track of 28 days cycles in affectivity in each others social media content, statistics on emotional expressions tells us new things.
- Depression, ethnicity etc. detection used Insurances.
- Creating a framework for detecting protest on chinese social media.

Agenda

Brush-up session

String operations and regular expressions

- Best practices for developing Regular Expressions.
  - These are crucial to text cleaning (and scraping which is not covered), and simple information extraction without heavy models. E.g. matching prices, phone numbers, emails or links in raw text.

Exploratory analysis: Visualization techniques

- Random samples of docs, sentences or contexts. Wordtree.
- WordClouds.

Exploratory analysis: Descriptives

- Distinctive words and Collocations
  - Chi2, Mann-Whitney, PMI
- Predictive words: modelbased explorations

BRUSH-UP SESSION

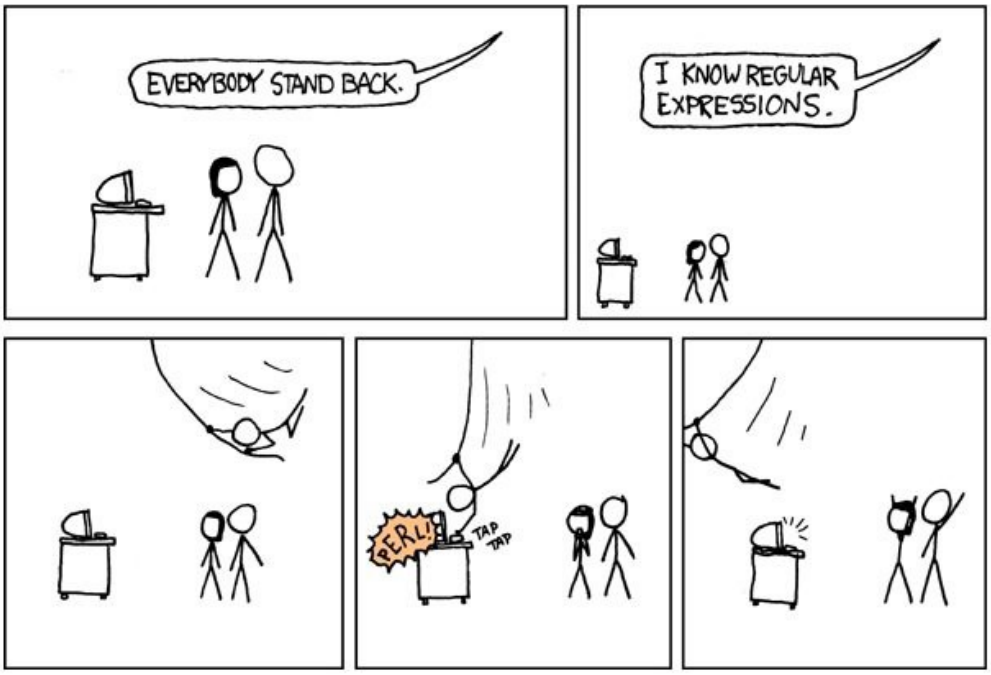
String-processing

of course you already now your basic string operations:

```
string.split
string.strip
string.replace
```

And sometimes this will be enough, but sometimes it is not.

Regular Expressions



Regex can be a little terrifying:

```
pattern = '(?:(?:\r\n)?[ \t])*(?:(?:(?:[^()<>@,;:
\\".\[\] \000-\031]+(?:(?:(?:\r\n)?[ \t])+|\Z|(?=
[\["()<>@,;:\\".\[\]]))|"(?:[^\"\r\\]|\\.|(?:
(?:\r\n)?[ \t]))*"(?:(?:\r\n)?[ \t])*)(?:\.(?:
(?:\r\n)?[ \t])*(?:[^()<>@,;:\\".\[\] \000-\031]+
(?:(?:(?:\r\n)?[ \t])+|\Z|(?=[\["()<>@,;:
\\".\[\]]))|"(?:[^\"\r\\]|\\.|(?:(?:\r\n)?[ \t]))*"
(?:(?:\r\n)?[ \t])*))*@(?:(?:\r\n)?[ \t])*(?:
[^()<>@,;:\\".\[\] \000-\031]+(?:(?:(?:\r\n)?[
\t])+|\Z|(?=[\["()<>@,;:\\".\[\]]))|\[([^\[\]\r\\]|
\\.)*\](?:(?:\r\n)?[ \t])*)(?:\.(?:(?:\r\n)?[
\t])*(?:[^()<>@,;:\\".\[\] \000-\031]+(?:(?:
(?:\r\n)?[ \t])+|\Z|(?=[\["()<>@,;:\\".\[\]]))|
\[([^\[\]\r\\]|\\.)*\](?:(?:\r\n)?[ \t])*))*|(?:[^()
<>@,;:\\".\[\] \000-\031]+(?:(?:(?:\r\n)?[ \t])+|
\Z|(?=[\["()<>@,;:\\".\[\]]))|"(?:[^\"\r\\]|\\.|(?:
(?:\r\n)?[ \t]))*"(?:(?:\r\n)?[ \t])*)*\<(?:(?:
\r\n)?[ \t])*(?:@(?:[^()<>@,;:\\".\[\] \000-\031]+
(?:(?:(?:\r\n)?[ \t])+|\Z|(?=[\["()<>@,;:
\\".\[\]]))|\[([^\[\]\r\\]|\\.)*\](?:(?:\r\n)?[
\t])*)(?:\.(?:(?:\r\n)?[ \t])*(?:[^()<>@,;:\\".\[\]
\000-\031]+(?:(?:(?:\r\n)?[ \t])+|\Z|(?=[\["()<>@,;:
\\".\[\]]))|\[([^\[\]\r\\]|\\.)*\](?:(?:\r\n)?[
\t])*))*(?:,@(?:(?:\r\n)?[ \t])*(?:[^()<>@,;:
\\".\[\] \000-\031]+(?:(?:(?:\r\n)?[ \t])+|\Z|(?=
[\["()<>@,;:\\".\[\]]))|\[([^\[\]\r\\]|\\.)*\](?:(?:
\r\n)?[ \t])*)(?:\.(?:(?:\r\n)?[ \t])*(?:[^()<>@,;:
\\".\[\] \000-\031]+(?:(?:(?:\r\n)?[ \t])+|\Z|(?=
[\["()<>@,;:\\".\[\]]))|\[([^\[\]\r\\]|\\.)*\](?:(?:
\r\n)?[ \t])*))*)*:(?:(?:\r\n)?[ \t])*)?(?:[^()
<>@,;:\\".\[\] \000-\031]+(?:(?:(?:\r\n)?[ \t])+|
\Z|(?=[\["()<>@,;:\\".\[\]]))|"(?:[^\"\r\\]|\\.|(?:
(?:\r\n)?[ \t]))*"(?:(?:\r\n)?[ \t])*)(?:\.(?:
```

Regular expressions (1)

Central to datacleaning and basic information extraction

Central to Understanding Tokenization of documents into words.

- Extract currency and amount from raw text: $ 20, 10.000 dollars 10,000 £
- email addresses: here you want to design a pattern (as above), that captures only the uses of @ within an email.
- urls. Here you are trying to define all the different ways of writing urls (https, http, no http).
- Dates. Again many variations: 17th of June 2017, 06/17/17 or 17. June 17
- addresses,
- phone numbers: 8888888 or 88 88 88 88 or +45 88 88 88 88,
- emojiies in text. Capturing all the different ways of expressing smiley faces with one regular expression.

Regular expressions (2)

Cheat-sheet

Regular Expression syntax

Ressources: Best way to learn is to practice, and with interactive examples. Two good ressources are here:

- Community and interactive playground here (http://regexr.com/)
- Interactive tutorial here (https://regexone.com/)
- or you can use your notebook.

Lookup all special characters here (https://www.regular-expressions.info/refquick.html)

- + = 1 or more times -- e.g. "a+" will match: "a", and "aaa"
- * = 0 or more times -- e.g. "ba*" will match: "b", and "ba", and "baaa"
- {3} = exactly three times --- e.g. "ba{3}" will match "baaa", but not "baa"
- ? = once or none
- \ = escape character, used to find characters that has special meaning with regex: e.g. + *
- [] = allows you to define a set of characters
- ^ = applied within a set, it becomes the inverse of the set defined. Applied outside a set it entails the beginning of a string. $ entails the end of a string.
- . = any characters except line break
- | = or statement. -- e.g. a|b means find characters a or b.
- \d = digits
- \D = any-non-digits.
- \s = whitespace-separator

Sequences

You will need the following for exercises tomorrow

Greedy but minimal matcher

Example.

- http.+.com - matches all characters between a "http" pattern and ".com" pattern.
- http.+?.com - matches all characters between a "http" pattern and the first ".com" pattern.

You will need the following for later

Capture groups

Used as a marking strategy in NLP.

```
"I HATE YOU" is translated into --> "__allcaps__ i hate you"
```

Example extracting currency and amount

```
'\[$£\](\d)+'
```

Capture groups:

```
r'\1 \2'
```

\1 is the specific currency.

\2 is the amount

Regular Expressions (3)

Best Practices for developing high quality expressions

You will use the following in the Exercises

I developed a small module for developing regular expressions in a systematic manner.

Just run the following piece of code to download, save and import the module.

```
# download module
url = 'https://raw.githubusercontent.com/snorreralund/explore_re
gex/master/explore_regex.py'
response = requests.get(url)
# write script to your folder to create a locate module
with open('explore_regex.py','w') as f:
    f.write(response.text)
# import local module
import explore_regex as e_re
```

Define, Inspect, Refine, Repeat (1)

- Learning and Exploring
- Balancing Precision and Recall.
    - Too narrow but right or too greedy but often wrong.
- Inspections and Re-Iteration

Define, Inspect, Refine, Repeat (2)

Workflow

1. Make a randomly sampled set of texts to test on.
    - Make sure it is large enough to capture even rare variations (but also not too big for computational reasons)
2. Start broad: e.g. $ or \d+
    - notice the no. matches of this greedy match, to get a sense of the recall.
    - sample with context to see variations.
3. Develop one step at a time
    - follow the no. of matches.
    - and keep inspecting with context.
4. Explore differences between patterns.
    - Soft and Hard difference.
        - e.g. Soft match between matches from the string `"$100.000,00"` $100 matches $100.000,

```
import explore_regex as e_re
eregex = e_re.ExploreRegex(textsample)
### Use the functions
    .explore_pattern(pattern)
    .explore_difference(pattern1, pattern2)
    .report()
```

In [22]:

```
import requests
url ='https://github.com/abjer/tsds'
text_sample = requests.get(url).text
import explore_regex as e_re
eregex = e_re.ExploreRegex(text_sample)
```

In [ ]:
```
#patterns = ['http','https','https?','\.com','http']
eregex.explore_pattern('http')
```

Exploratory analysis of text data

"As with any modelling effort, understanding your data is central"

- Quality of the data
- Content and variations in the data.
- Summarization and Visualization Techniques for text data

For Assignments

Visualizations provide transparancy to the analysis.

Give the reader the ability to look into the data and the problems.

Exploratory analysis of text data

Visualization (1)

- Manual inspections
  - Random samples of documents, sentences, or contexts.
- WordTree

In [8]:
```
from IPython.display import IFrame
url = 'https://www.jasondavies.com/wordtree/?source=obama.inaugur
ation.2013.txt&prefix=We&phrase-line=0&reverse=0'
#IFrame(url,800,800)
```

Exploratory analysis of text data

Visualization (2)

- WordClouds (you will play with this)

In [19]:
```
from IPython.display import Image
Image(filename = 'word_cloud_R_Python.png')
```

Out[19]:

Exploratory analysis of text data

Descriptives (1)

Visualizing basic descriptive distributions in your data using standard methods:

- Distributions of word frequencies (long tailed zipf distribution)
- No. characters, sentences, commas, words etc. pr. document.
- Average word lengths, lix number, allcaps characters.

Simple Statistical Summarization

- Average word lengths, lix number
- Lexical diversity distribution accross
- Entropy

Exploratory analysis of text data

Descriptives(2)

Distinctive Terms

Learning about Classes or Subgroups in data

X^2 - chisquared

The statistical co-occurence strengths of two or more events. Used to find Word-Phrases of two words being co-located next to each other (e.g. New York), or a word occurring when a certain group of people speak.

Consider the contingency table O:

O	**bacon**	$\neg bacon$
vegan	10	990
$\neg vegan$	2500	10000

X^2 express the difference between the co-occurence we observe and what we would have expected to see if two events independent, relative to the latter. I.e. how many times more prevalent is the co-occurrence we observe than what we would expect if they were independent.

$X^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$ where i ranges over rows of the table, j ranges over columns, O_{ij} is the observed value for cell (i, j) and $E_i j$ is the expected value.

and $E_{ij} = N(p_i * p_j)$

In [15]:
```
import pandas as pd
import json
recipe_df = pd.read_csv('/home/snorre/Dropbox/Forskning/PhD/under
visning/recipe_data/epi_r.csv')
#recipes = json.load(open('/home/snorre/Dropbox/Forskning/PhD/und
ervisning/recipe_data/recipe_data/full_format_recipes.json','r'))
import nltk
tokens = recipe_df.title.apply(lambda x: [i.lower() for i in nltk
.word_tokenize(x)])
from collections import Counter
veg_count = Counter([i for doc in tokens[recipe_df.vegan==1] for
i in doc])
nonveg_count = Counter([i for doc in tokens[recipe_df.vegan==0] f
or i in doc])
veg_N = sum(veg_count.values())
nonveg_N = sum(nonveg_count.values())
N = veg_N+nonveg_N

nonveg_freq = nonveg_N/N
veg_freq = veg_N/N
for ingr in ['salad','sauce','bacon']:

    # observed values
    v_c = veg_count[ingr]
    nv_c = nonveg_count[ingr]

    # ingredient count, both vegan and non vegan
    ingr_c = v_c+nv_c

    # expected counts given independence.
    exp_ingr_v = ingr_c*veg_freq
    exp_ingr_nv = ingr_c*nonveg_freq

    # substract observed from expected and divide by expected
    chi2_v = ((v_c-exp_ingr_v)**2)/exp_ingr_v
    chi2_nv = ((nv_c-exp_ingr_nv)**2)/exp_ingr_nv
    # sum over values to get values from all
    chi2 = chi2_v+chi2_nv
    # get direction
    direction = (v_c-exp_ingr_v)>0
    if direction:
        direction = 'more'
```