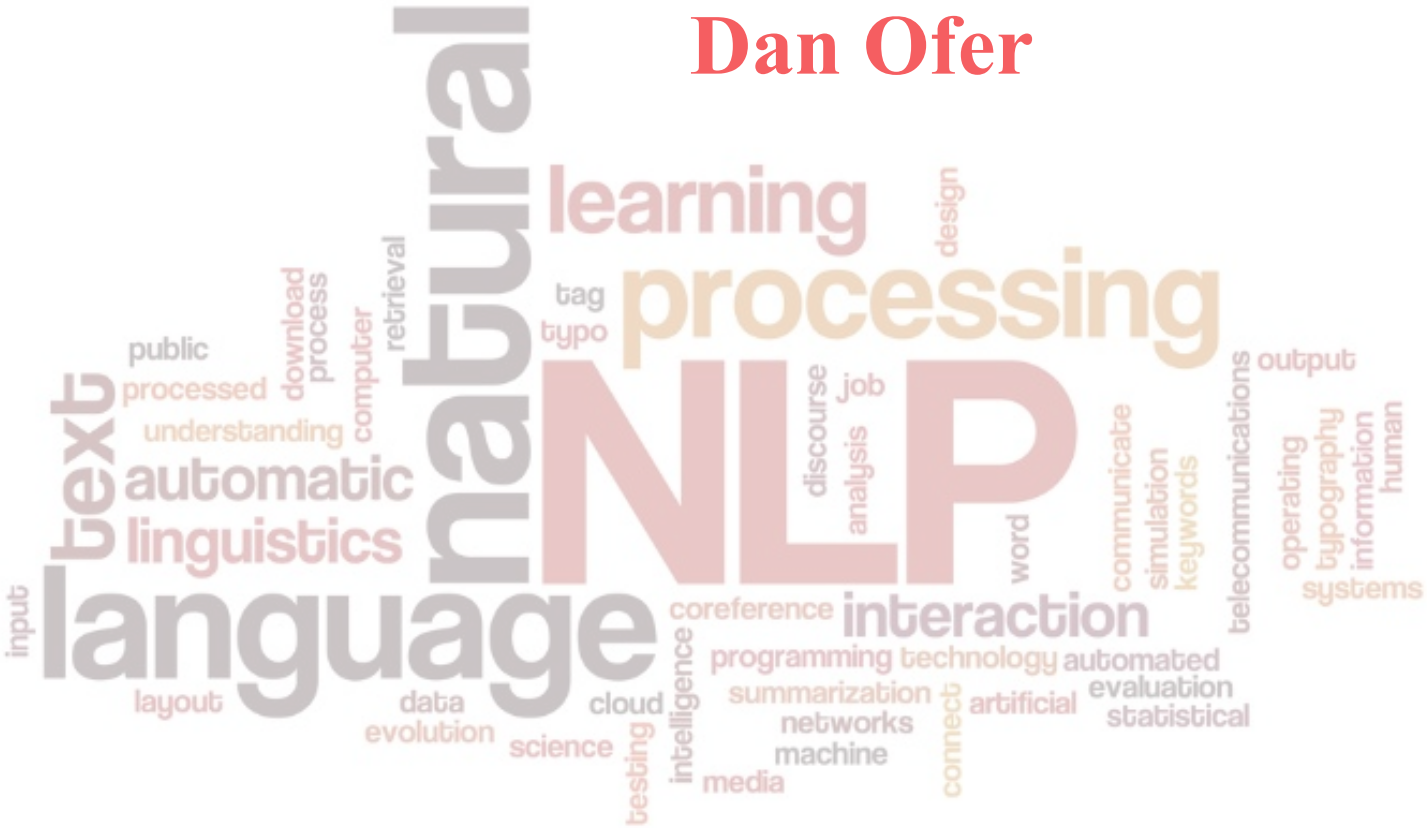
NLP (Natural Language Processing) 101

Dan Ofer

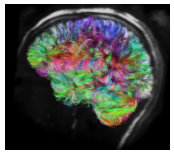


Dan Ofer
Data Scientist @ SparkBeyond

THE HEBREW UNIVERSITY OF JERUSALEM

MsC: Neuroscience & Bioinformatics
- HUJI @ M. Linial: Neuropeptides & Protein feature engineering.

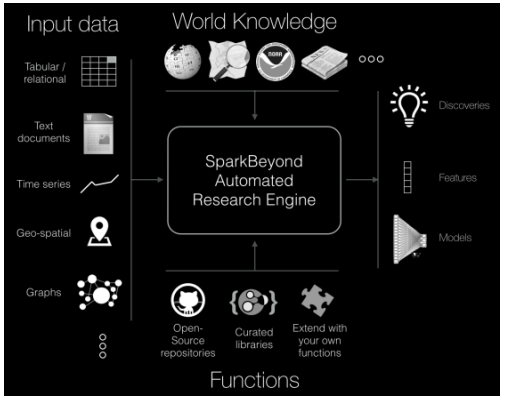
```
F. Corticotropin-releasing hormone (CRH)-type precursor (ArCRHP)
MNDLQRLILLVSLGTFALLLCLPACTEAQPLGLFKFEYDDLLDPSFEADDPRNPRRLSRQQILRRL
AMSRSGSGPGYTIPRKRQGLSVSPIFPIQRIRLNAIERDRQDQVDQAEANQGLFQIAGRKR
```

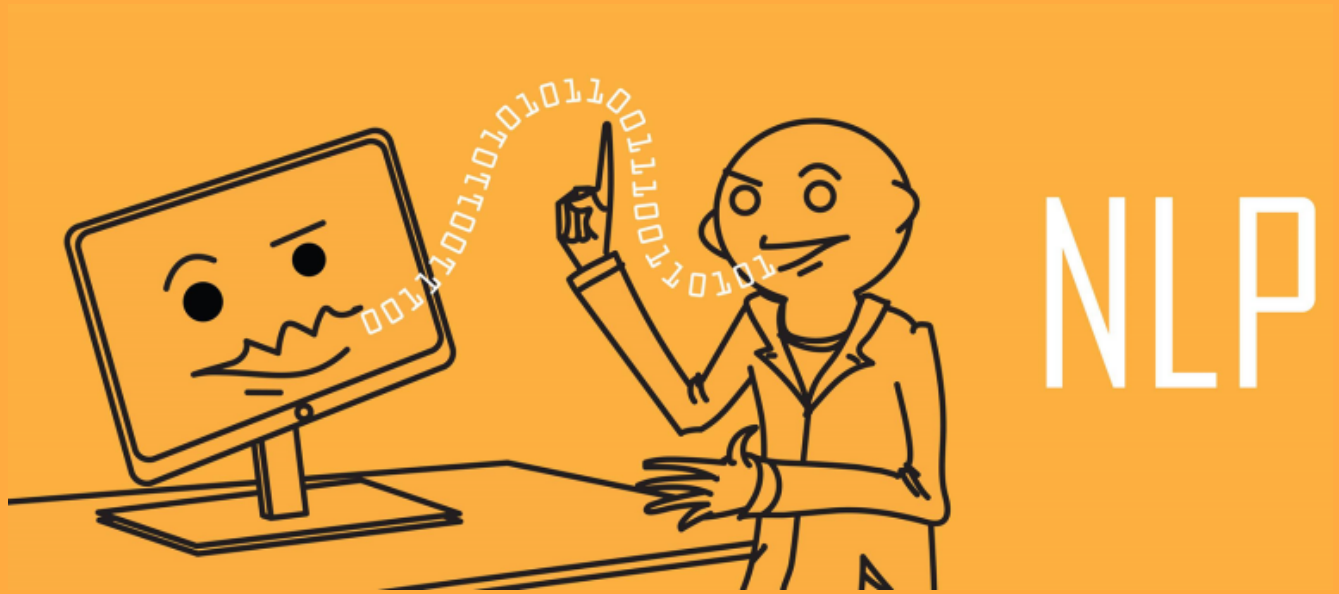


BsC: Psychobiology

Army: Translation = תרגום

dan@sparkbeyond.com





Natural Language Processing <=> Computer Science + AI + Computational Linguistics

NLP systems: “understand” natural (i.e unstructured) language in order to infer intent/tasks.

why 1S nlp 50 h4rd

- Ambiguity
- Unstructured (relatively)

ברחתי מהאיש עם המספריים

I ran away from the man with scissors

i ran away from the man with the scissors Edit

ברחתי מהאיש עם מספריים

Why is NLP hard?

I shot an elephant in my pajamas.

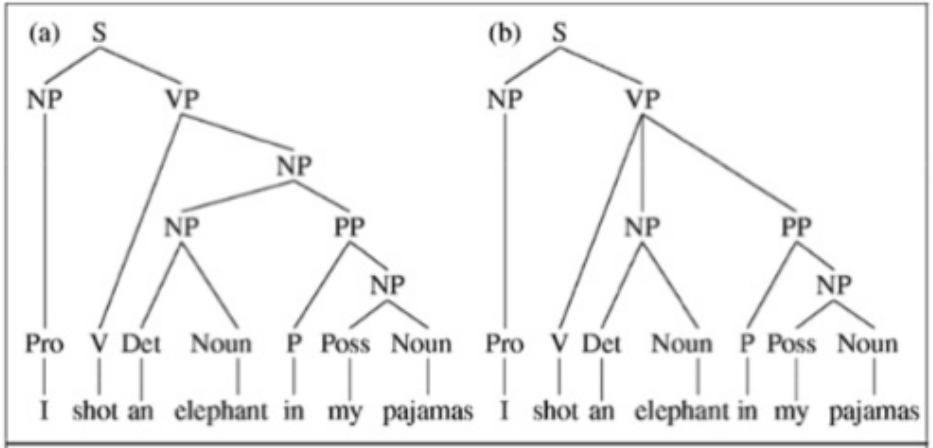


Figure 10.11 Two parse trees for an ambiguous sentence. Parse (a) corresponds to the humorous reading in which the elephant is in the pajamas, parse (b) to the reading in which Captain Spaulding did the shooting in his pajamas.

WIKIWORLD by Greg Williams

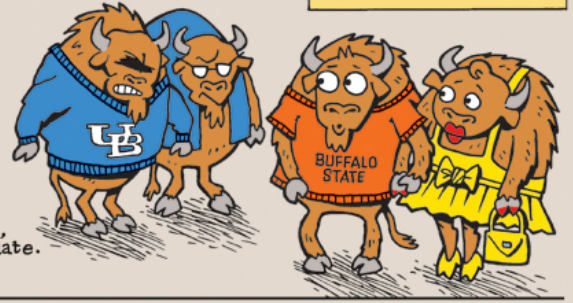
Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo.

is a grammatically correct sentence used as an example of how homonyms and homophones can be used to create complicated constructs. The sentence is unpunctuated and uses three different readings of the word "buffalo." In order of their first use, these are:

- The city of Buffalo, New York.
- The animal "buffalo," in the plural (equivalent to "buffaloes"), in order to avoid articles.
- The verb "buffalo," meaning to confuse, deceive or intimidate.

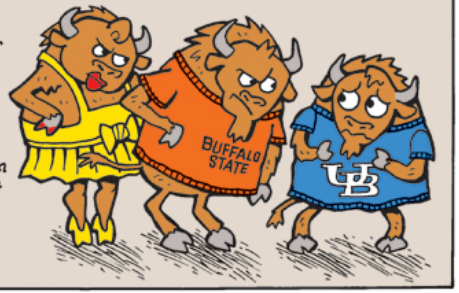
Homonym = a word form that has two or more distinct meanings

Homophone = a word which is pronounced the same as another word but differs in meaning



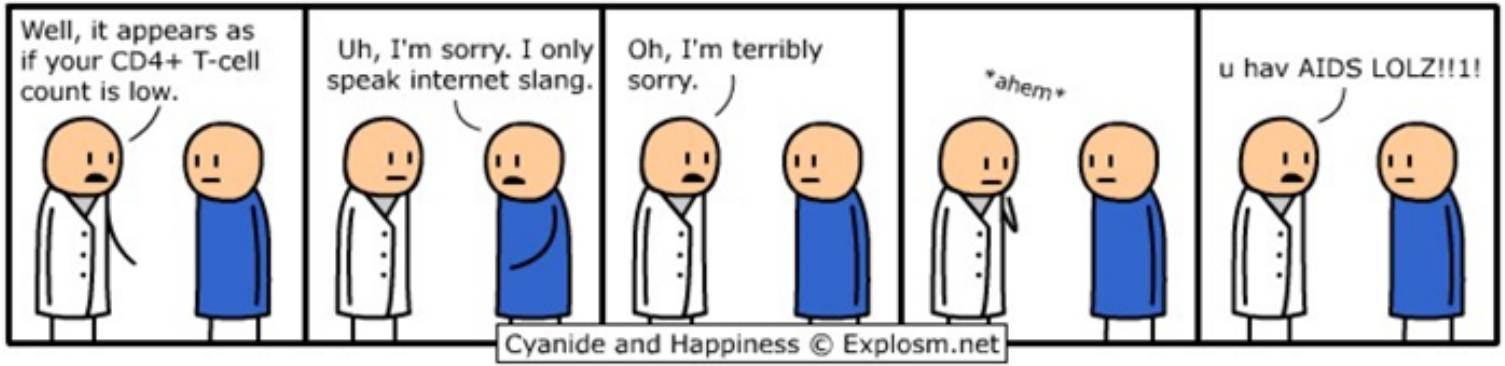
Substituting the synonym "bison" for "buffalo" (animal), "bully" for "buffalo" (verb) and leaving "Buffalo" to mean the city, yields:

Buffalo bison, whom other Buffalo bison bully, themselves bully Buffalo bison.



Text excerpted from the Wikipedia articles *Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo buffalo.*, *Homonym* and *Homophone*. 26 March 2007

Non-Standard Language



Also: neologisms, complex entity names, phrasal verbs/idioms

Dan Jurafsky

Why else is natural language understanding difficult?

non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either♥

segmentation issues

the New York-New Haven Railroad
the New York-New Haven Railroad

idioms

dark horse
get cold feet
lose face
throw in the towel

neologisms

unfriend
Retweet
bromance

world knowledge

Mary and Sue are sisters.
Mary and Sue are mothers.

tricky entity names

Where is *A Bug's Life* playing …
Let It Be was recorded …
… a mutation on the *for* gene …

English is easy..

- **German**: Donaudampfschiffahrtsgesellschaftskapitän (5 “words”)
- **Chinese**: 50,000 different characters.
- **Japanese**: 3 writing systems
- **Thai**: Ambiguous word boundaries and sentence concepts
- **Slavic**: Different word forms depending on gender, case
  - **Hebrew**: ambiguous Roots (e.g. שלם), lots more..

I‘m sorry Dave,
I‘m afraid that I can‘t do that.

What can NLP do?

Common NLP Tasks

Easy

- Tokenization
  - Predict Word-level Labels/class
- Part-of-Speech Tagging. (Verb, noun, adjective…)
- Named Entity Recognition (NER)

Medium

- Predict a text's class
  - Sentiment Analysis
- Syntactic (syntax) Parsing
- Word Sense Disambiguation
- Topic Modeling (LDA)
  - Audio-> Text..

Hard

- Machine Translation
- Text Generation
- Summarization
- Question Answering
- Conversational Interfaces
  - Understanding. (Grade essays)

NLP ‘annotations’/tasks

Question Answering

Siri: ‘When does Shabat end?’

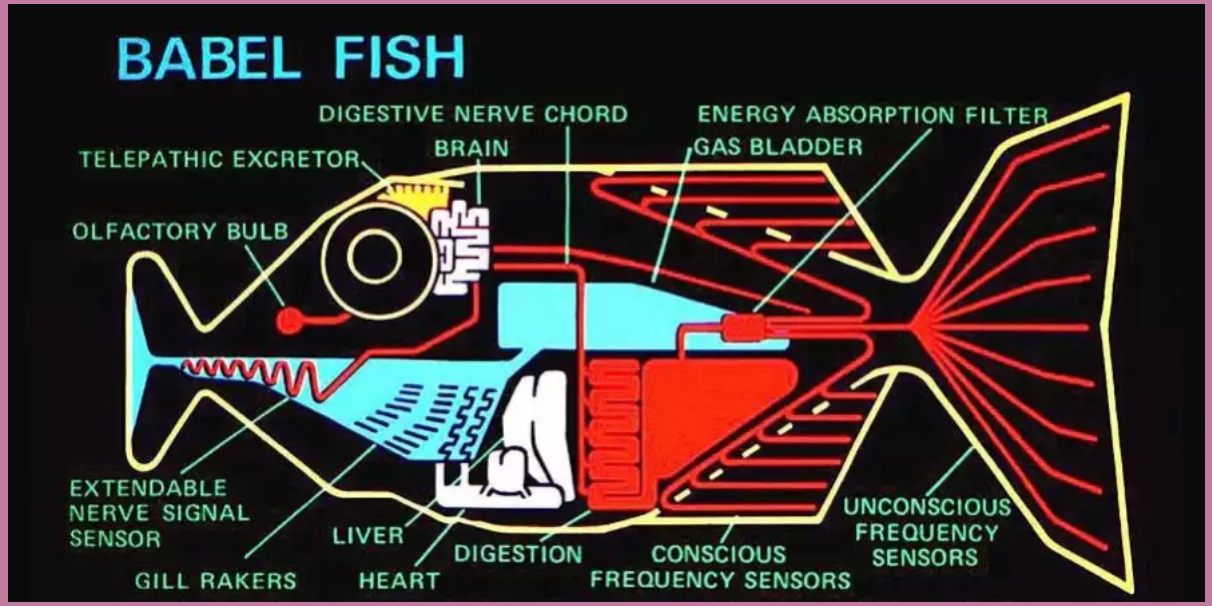
Sentiment Analysis.

‘The movie was great, in the same way the Matrix sequels were.’

‘Help help help!!!’

ns/ca

Translation



Quiero ir a la playa más bonita.

I want to go to the beach more pretty.

Many more uses...

- Automatic summarization
- Coreference resolution
- Discourse analysis
- Machine translation
- Morphological segmentation
- Named entity recognition (NER)
- Natural language generation
- Word sense disambiguation
- Relationship extraction
- Speech processing
- Part-of-speech tagging
- sentence boundary disambiguation
- Sentiment analysis
- Optical character recognition (OCR)
- Question answering
- Parsing
- Word segmentation
- Natural language understanding
- Information retrieval (IR)
- Speech recognition
- Topic segmentation and recognition
- Speech segmentation
- Information extraction (IE)

ML, NLP & You: Feature Engineering

Getting started in NLP & ML

Basics First. Deep Recursive, Recurrent network with POS vectors & knowledge ontologies later.

Bag of Words + preprocessing = can do a LOT, fast!

Good libraries to know: Gensim, spaCy, CoreNLP, TextBlob, NLTK, Polyglot, Textacy, tm (R)..

Can have many components, can be building blocks or tasks themselves..

The Classic: Bag of Words (BOW)

Count the Words!

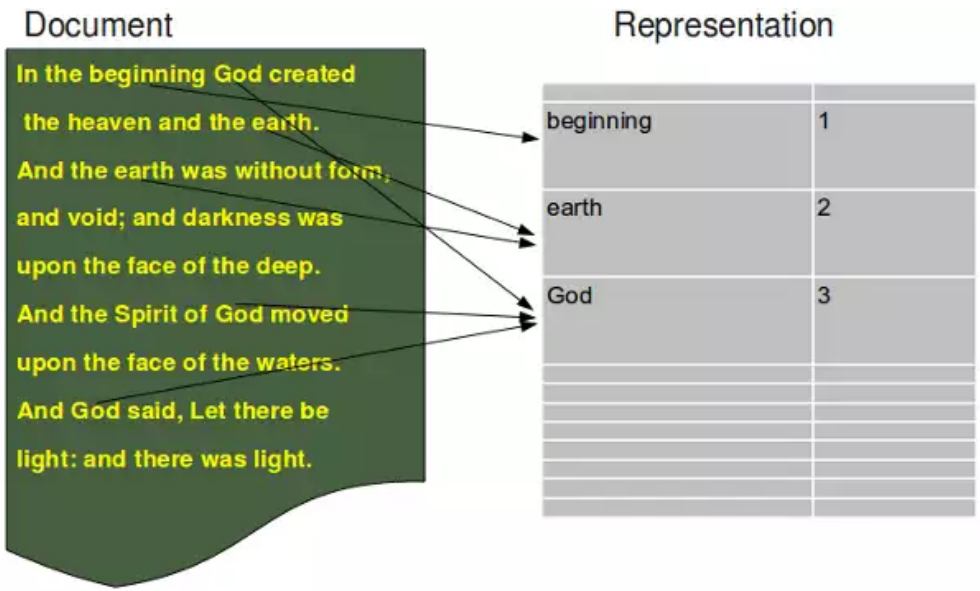
Sparse, one-hot encoded vector

Dimension = vocab size

Preprocessing crucial!:

Tokenize

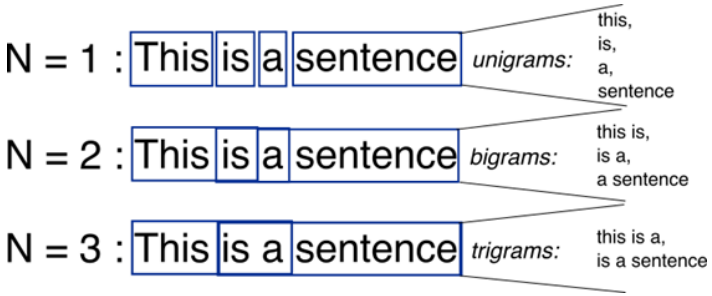
Stem



n-grams

Gets big FAST.

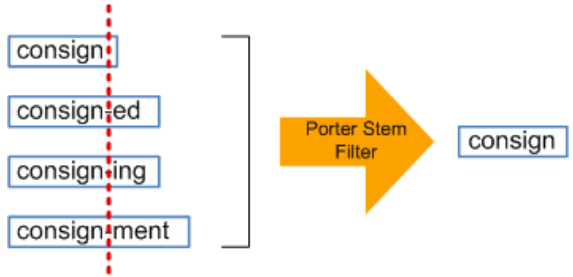
BONUS: K-mers, skipgrams..



Basics of Preprocessing: Extract, tokenize, Stem, Lemmatize...

Stem:

"Chop off the end". Rule based.



Lemmatize: 'Basic form/meaning of the word in the dictionary'

Eat = Consume. Feebleminded = retarded…

run, runs, ran, running -> run

Requires dictionary (e.g. WordNet) & POS tags

- "Produced":
  - lemma ="produce".
  - Stem ="produc-".

Getting started in NLP ML: The Classic TF

TF, TF-IDF: **t**erm **f**requency–**i**nverse **d**ocument **f**requency.

Top method in information retrieval!

Fast, effective.

Still loses word-ordering.

IDF: ‘Unique’/informative words = better than simply common words.

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

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

TF-IDF Simpsons' Summaries

Episode title	tf-idf summary
Bart the Genius	kwyjibo
Homer's Odyssey	nuclear energy
Bart the General	sound off
The Telltale Head	jebediah
Krusty Gets Busted	krusty
Bart Gets an "F"	blah blah blah blah
Simpson and Delilah Treehouse of Horror Last Exit to Springfield	Dimoxinil Nevermore Dental Plan

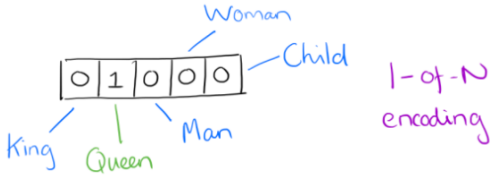

WE NEED TO GO
DEEPER
memegenerator.net

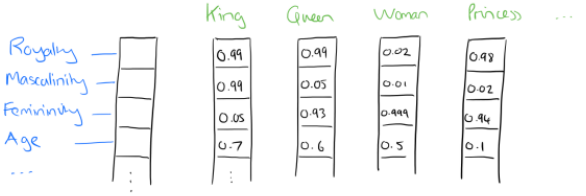
Distributed Word Embeddings AKA Word2Vec

- OHE/ Bag of words model is very high-dimensional, sparse, doesn't naturally capture similarities.
- **Idea**: represent words as **dense vectors**

~~[0 1 0 0 0 0 0 0 0 0]~~

[0.315 0.136 0.831]

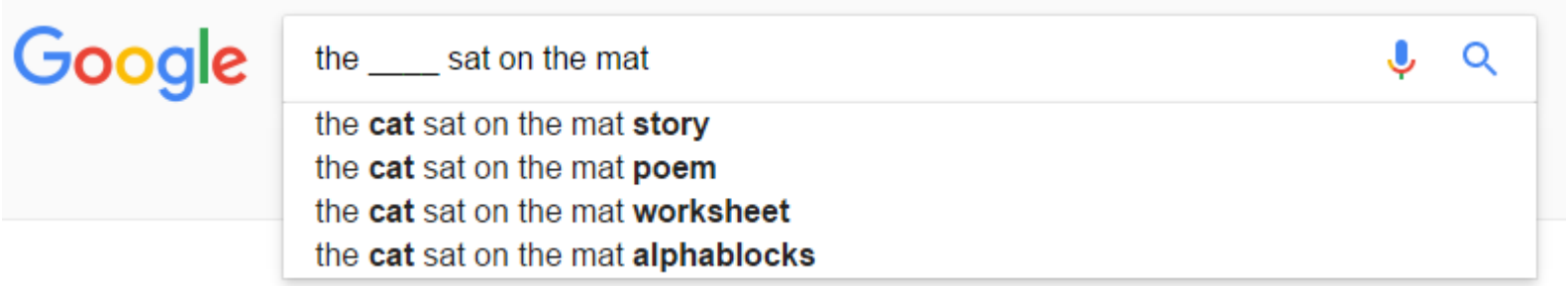




Many algorithms: Word2Vec, GloVe, FastText, WordRank, etc'

Effective even without DL models, Seq2seq, LSTMs, etc'

Word Vectors: The Big Idea



The ____ sat on the mat

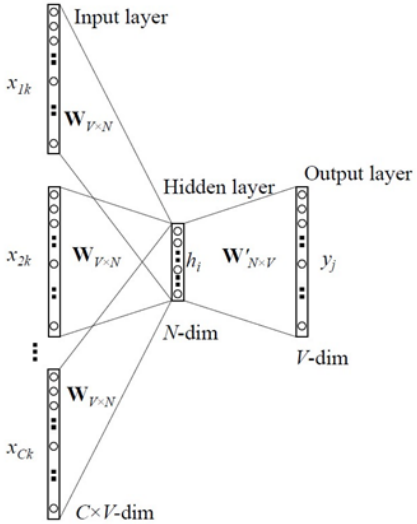
Cat? Dog?

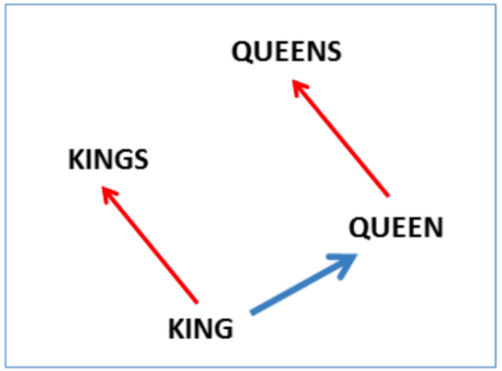
Entomophagy (eating of insects by people)?

Supercalifragilisticexpialidocious?

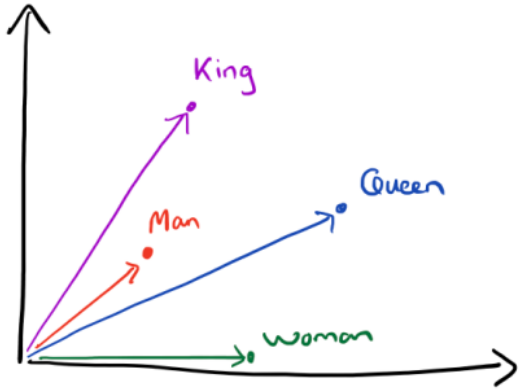
Suoicodilaipxecitsiligarfilacrepus ?

Word2Vec Embedding:





- **CBOW: P**redict center word given context (Surrounding words).
- **Skip-gram version**: predict context from center word
  - W2V tricks: Negative subsampling, speed (Shallow), # data, phrase coallocation..



Word Vectors


```
In [16]: model.most_similar('king')
Out[16]:
[(u'throne', 0.7332440614700317),
 (u'kings', 0.7032474875450134),
 (u'crowned', 0.7003846764564514),
 (u'monarch', 0.6924914717674255),
 (u'prince', 0.6895323395729065),
 (u'eochaid', 0.6760289669036865),
 (u'son', 0.6715421676635742),
 (u'reigned', 0.6627429127693176),
 (u'vii', 0.6580543518066406),
 (u'reign', 0.6530766487121582)]
```

WordRank

```
In [26]: model.most_similar('king')
Out[26]:
[(u'eochaid', 0.8079677820205688),
 (u'canute', 0.792285144329071),
 (u'mormaer', 0.7795065641403198),
 (u'capet', 0.7787382006645203),
 (u'bouillon', 0.7762842178344727),
 (u'alpin', 0.7752912044525146),
 (u'godwinson', 0.7732000946998596),
 (u'gundahar', 0.771564781665802),
 (u'conradin', 0.7687084078788757),
 (u'chatillon', 0.7666929960250854)]
```

Word2Vec

```
In [32]: model.most_similar('king')
Out[32]:
[(u'thrones', 0.7961102724075317),
 (u'son', 0.7955597639083862),
 (u'godred', 0.7742120027542114),
 (u'therion', 0.7590411901473999),
 (u'pretender', 0.7583349347114563),
 (u'prince', 0.7535479664802551),
 (u'thron', 0.7528668642044067),
 (u'throne', 0.7479698657989502),
 (u'godfred', 0.7392275333404541),
 (u'pretended', 0.7372602820396423)]
```

FastText

Tokenization & Stemming

```
In [1]: import nltk

In [2]: nltk.download(["punkt", "averaged_perceptron_tagger", "treebank",
                       "maxent_ne_chunker", "words", "wordnet"])

...

In [3]: text = open("data/clean/asoiaf01.txt").read()

In [4]: sentences = nltk.tokenize.sent_tokenize(text)

In [5]: sentences[5313]
Out[5]: 'Cersei Lannister regarded him suspiciously.'

In [6]: sentences = [nltk.tokenize.word_tokenize(sentence) for sentence in sentences]

In [7]: sentences[5313]
Out[7]: ['Cersei', 'Lannister', 'regarded', 'him', 'suspiciously', '.']

In [8]: stemmer = nltk.stem.SnowballStemmer(language="english")
        [stemmer.stem(word) for word in sentences[5313]]
Out[8]: ['cersei', 'lannist', 'regard', 'him', 'suspici', '.']
```

POS/NER Tagging

```
In [10]: tagged_sentence = nltk.pos_tag(sentences[5313])

In [11]: tagged_sentence

Out[11]: [('Cersei', 'NNP'),
          ('Lannister', 'NNP'),
          ('regarded', 'VBD'),
          ('him', 'PRP'),
          ('suspiciously', 'RB'),
          ('.', '.')]

In [12]: lemmatizer = nltk.stem.wordnet.WordNetLemmatizer()
         [lemmatizer.lemmatize(word, pos=penn_to_wn(tag)) for word, tag in tagged_sentence]

Out[12]: ['Cersei', 'Lannister', 'regard', 'him', 'suspiciously', '.']

In [13]: tree = nltk.ne_chunk(tagged_sentence)

In [14]: tree.draw()
```

Text Generation

"Jane hit June and then she [**fell**/*ran*]."

Old days: Markov. Today: LSTM (RNNs).

Character level VS Word level models.

Not just words: music, etc'.

Karpathy: The Unreasonable Effectiveness of Recurrent Neural Networks

HPMOR: https://www.facebook.com/notes/dan-ofer/harry-potter-char-rnn/10154441648502719

The End!

! Toda Raba!תודה רבה