Introduction to Natural Language Processing

Vered Shwartz

Bar-Ilan University, Israel

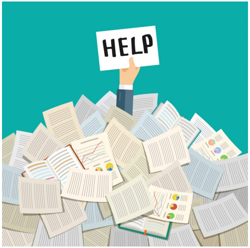
June 13, 2018

TMI: Too Much Information

Let's start with two facts:

TMI: Too Much Information

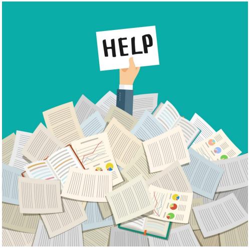
Let's start with two facts:



- 90% of the data in the world today has been created in the last two years.[1]

TMI: Too Much Information

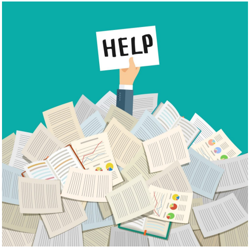
Let's start with two facts:



- 90% of the data in the world today has been created in the last two years.[1]
- Our attention span is now less than that of a goldfish,[2] and we almost never read through an article.[3]

TMI: Too Much Information

Let's start with two facts:

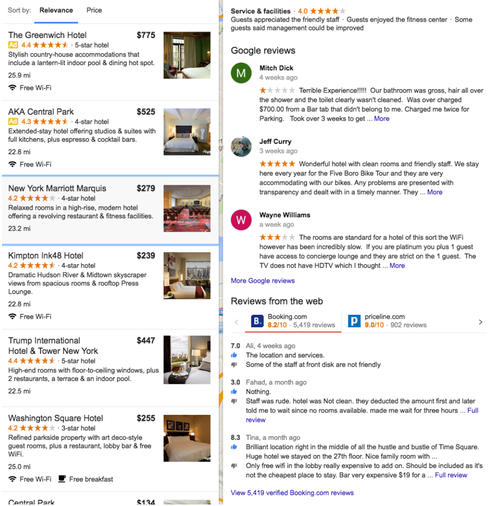


- 90% of the data in the world today has been created in the last two years.[1]
- Our attention span is now less than that of a goldfish,[2] and we almost never read through an article.[3]

[1] IBM, March 2012 (!)
[2] The Telegraph, March 2016
[3] Slate, March 2016

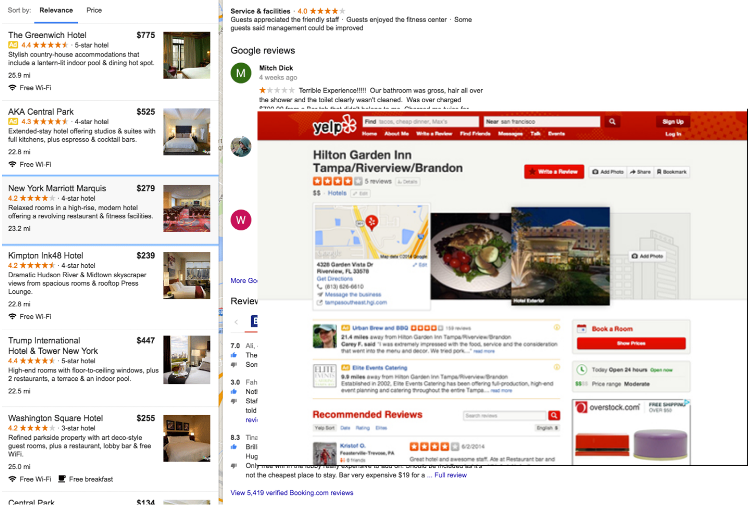
TMI: Too Much Information

Why is this a problem?



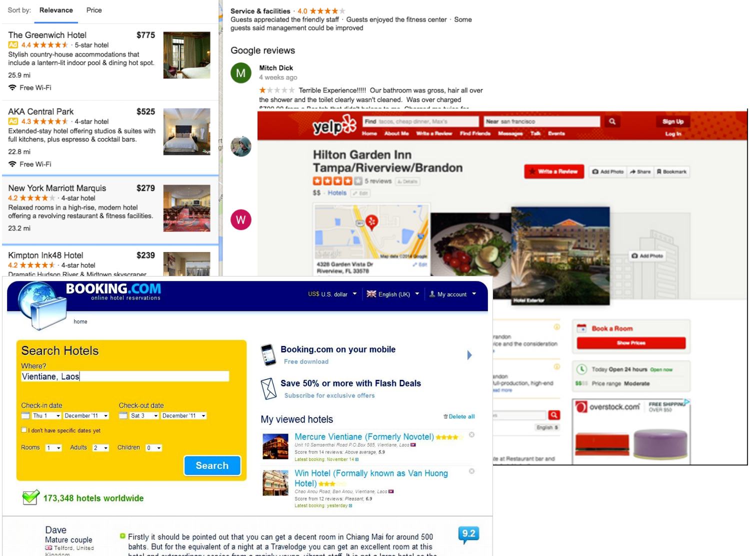
TMI: Too Much Information

Why is this a problem?

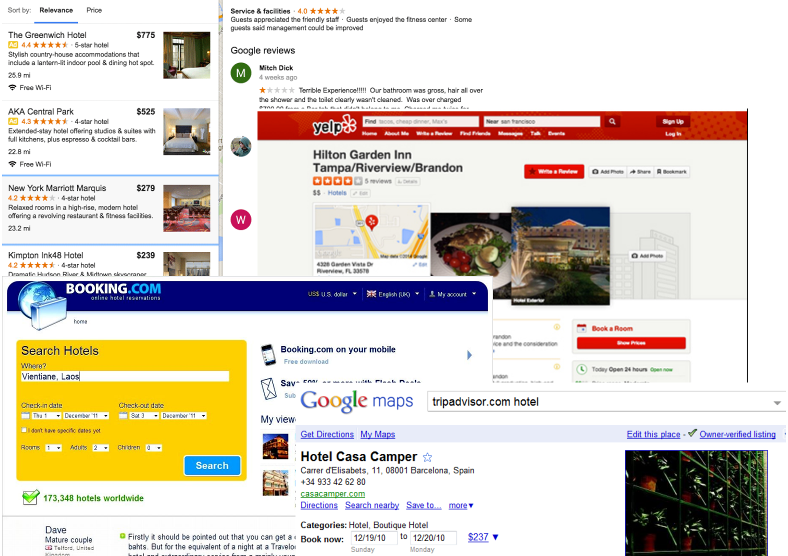


TMI: Too Much Information

Why is this a problem?

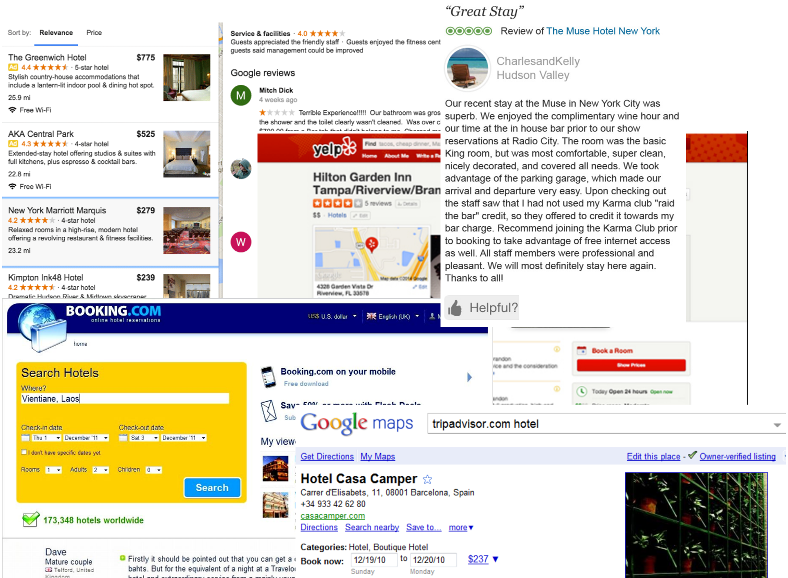


TMI: Too Much Information
Why is this a problem?



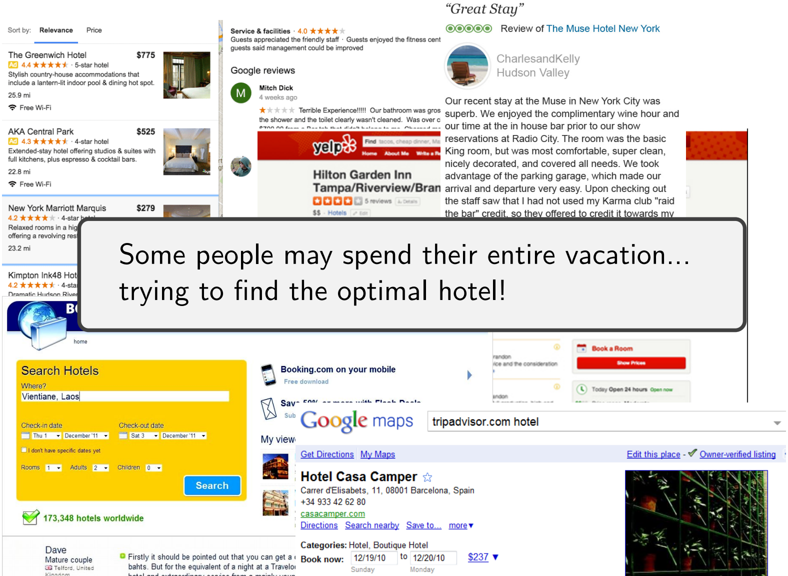
TMI: Too Much Information

Why is this a problem?



TMI: Too Much Information

Why is this a problem?



Some people may spend their entire vacation...
trying to find the optimal hotel!

Natural Language Processing to the rescue

We are working on automatic methods to...

- Summarize multiple long texts

Natural Language Processing to the rescue

We are working on automatic methods to...

- Summarize multiple long texts
- Answer questions based on texts

Natural Language Processing to the rescue

We are working on automatic methods to...

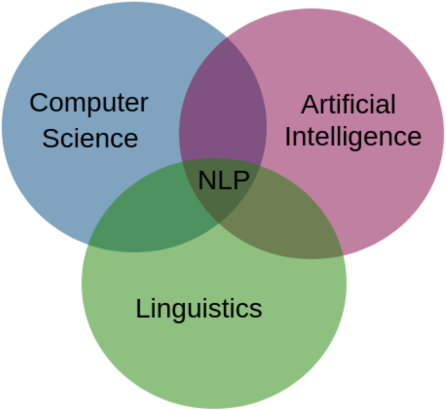
- Summarize multiple long texts
- Answer questions based on texts
- Identify the sentiment of texts (e.g. reviews)

Natural Language Processing to the rescue

We are working on automatic methods to...

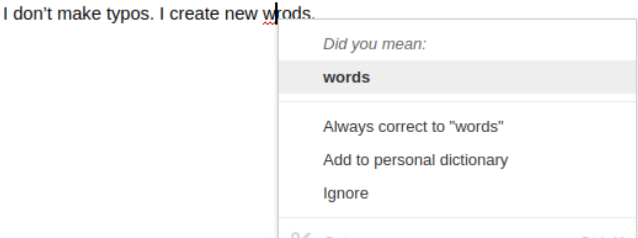
- Summarize multiple long texts
- Answer questions based on texts
- Identify the sentiment of texts (e.g. reviews)
- More...

What is Natural Language Processing (NLP)?



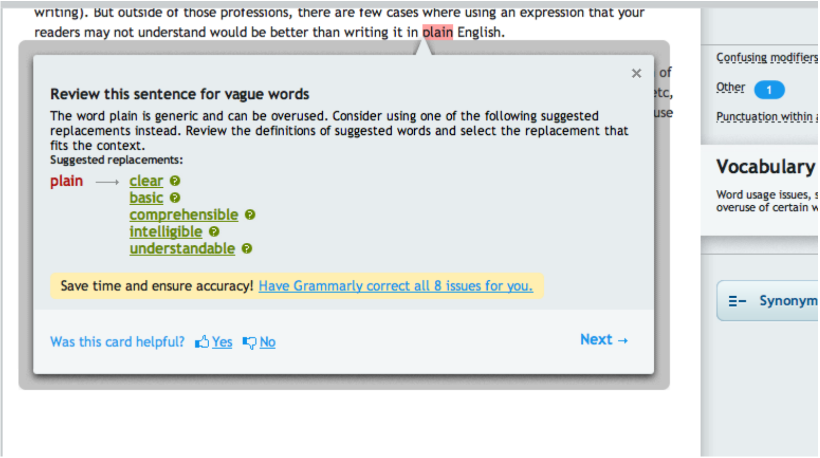
- **Goal:** for computers to “understand” and be able to communicate with people in natural languages (e.g. English)

NLP Applications are Everywhere
Spell Check

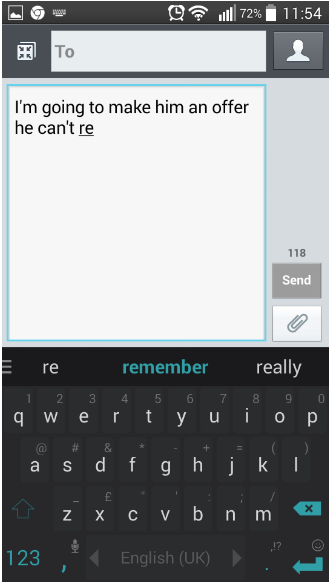


NLP Applications are Everywhere

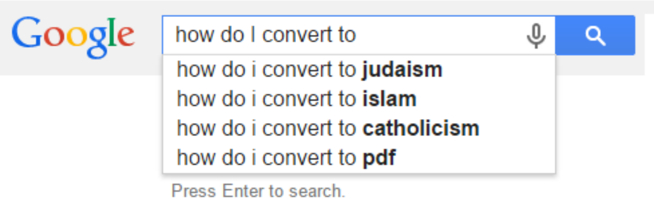
Grammar Correction



NLP Applications are Everywhere
Autocomplete

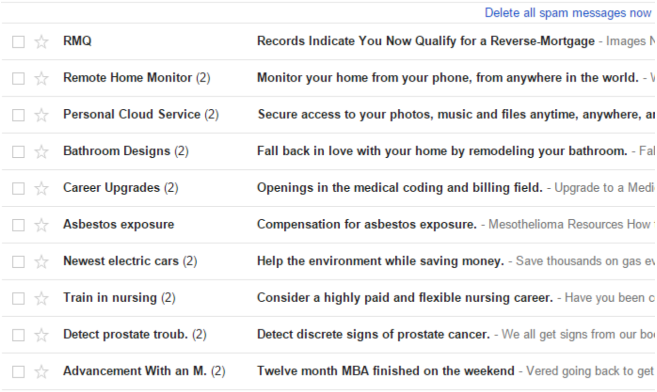


NLP Applications are Everywhere
Autocomplete



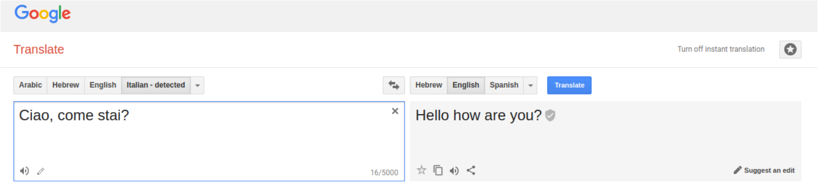
NLP Applications are Everywhere

Spam Detection



NLP Applications are Everywhere

Machine Translation



NLP Applications are Everywhere

Search Queries

Query

"Actors engaged in Scientology"

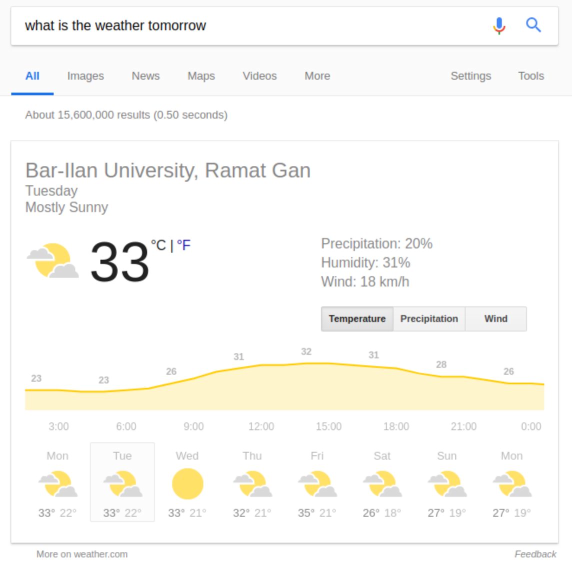
Results

Tom Cruise must ditch the vile cult of Scientology NOW before
www.dailymail.co.uk/.../PIERS-MORGAN-**Tom-Cruise**-ditch-vile-cult-**Scie**...
Apr 2, 2015 - PIERS MORGAN: I wanted to be **Tom Cruise**. As a fresh-faced, 21-year-old, I watched Top Gun a dozen times at my local movie theater in ...

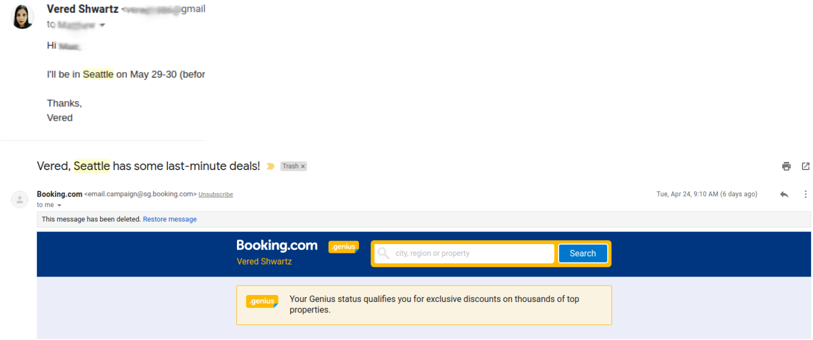
John Travolta Says Scientology Is A Target Because It ...
www.huffingtonpost.com/.../**john-travolta-scientology**-target_n_710268...
Apr 20, 2015 - During an interview with "Good Morning America" Monday, **John Travolta** was asked why there is so much intrigue and interest surrounding the ...

NLP Applications are Everywhere

Question Answering

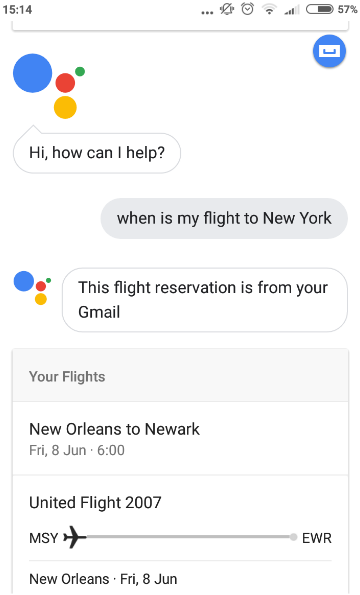


NLP Applications are Everywhere
Targeted Ads

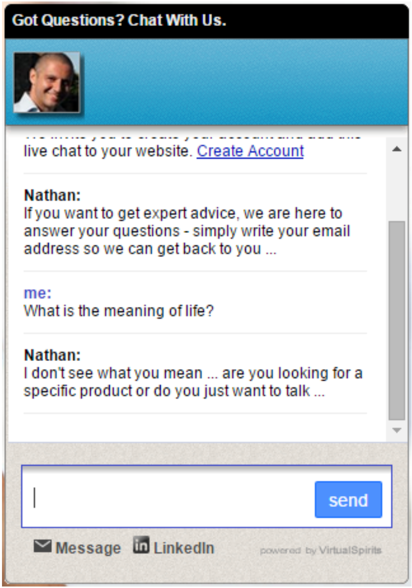


NLP Applications are Everywhere

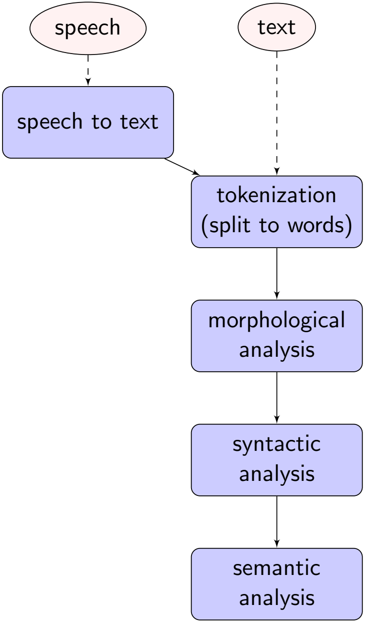
Personal Assistants



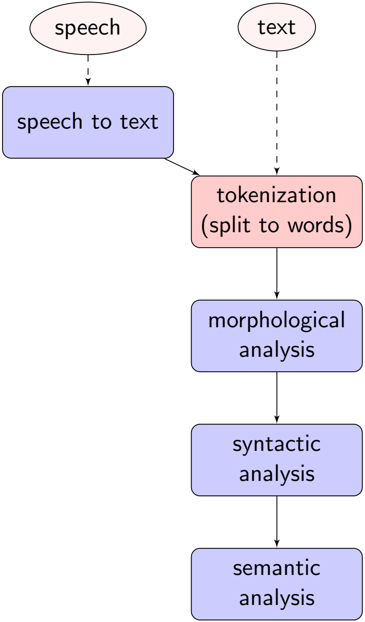
NLP Applications are Everywhere
Chatbots



Text Analysis Tasks



Text Analysis Tasks



Text Analysis Tasks

Tokenization

- Split text into a sequence of tokens ($\approx$ words)

Text Analysis Tasks

Tokenization

- Split text into a sequence of tokens ($\approx$ words)
- Naive approach: split sentences by period, words by spaces

Text Analysis Tasks

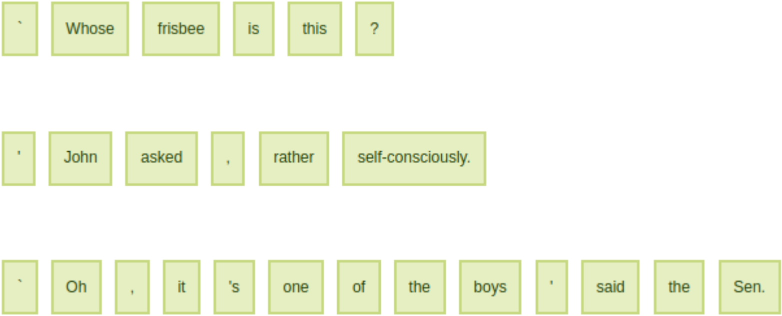
Tokenization

- Split text into a sequence of tokens ($\approx$ words)
- Naive approach: split sentences by period, words by spaces
- **How to tokenize this text?**
  ‘Whose frisbee is this?’ John asked, rather self-consciously.
  ‘Oh, it’s one of the boys’ said the Sen.

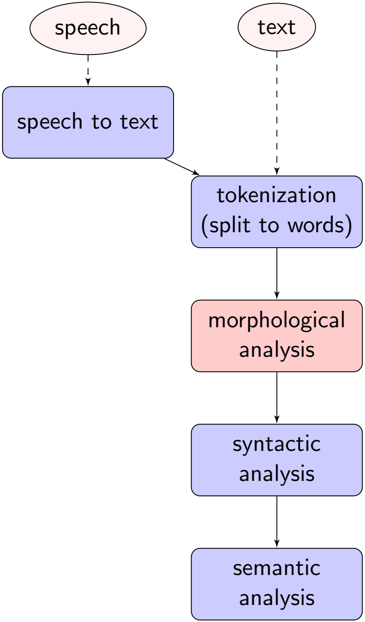
Text Analysis Tasks

Tokenization

- Split text into a sequence of tokens ($\approx$ words)
- Naive approach: split sentences by period, words by spaces
- **How to tokenize this text?**
  'Whose frisbee is this?' John asked, rather self-consciously. 'Oh, it's one of the boys' said the Sen.
- **(Optional) answer:**



Text Analysis Tasks



Text Analysis Tasks
Morphological Analysis

- Words are made from *morphemes*, smaller meaningful units

Text Analysis Tasks
Morphological Analysis

- Words are made from *morphemes*, smaller meaningful units
- Normally: base form + affixes

Text Analysis Tasks
Morphological Analysis

- Words are made from *morphemes*, smaller meaningful units
- Normally: base form + affixes
  - **Nouns** - plural form: dog**s**, suffix**es**, baby $\rightarrow$ bab**ies**

Text Analysis Tasks

Morphological Analysis

- Words are made from *morphemes*, smaller meaningful units
- Normally: base form + affixes
  - **Nouns** - plural form: dog**s**, suffix**es**, baby → bab**ies**
  - **Verbs** - tense: work**ed**, work**ing**, person: work**s**

Text Analysis Tasks
Morphological Analysis

- Words are made from *morphemes*, smaller meaningful units
- Normally: base form + affixes
  - **Nouns** - plural form: dog**s**, suffix**es**, baby → bab**ies**
  - **Verbs** - tense: work**ed**, work**ing**, person: work**s**
- Many irregularities... *"women and children begun running away as the wolves showed their teeth"*

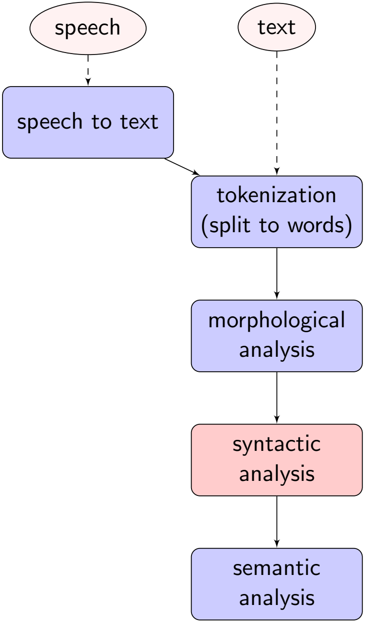
Text Analysis Tasks
Morphological Analysis

- Words are made from *morphemes*, smaller meaningful units
- Normally: base form + affixes
  - **Nouns** - plural form: dog**s**, suffix**es**, baby → bab**ies**
  - **Verbs** - tense: work**ed**, work**ing**, person: work**s**
- Many irregularities... *"women and children begun running away as the wolves showed their teeth"*
- Morphological analysis:
  - input: "am", output: "be" + 1 PERSON + PRESENT

Text Analysis Tasks
Morphological Analysis

- Words are made from *morphemes*, smaller meaningful units
- Normally: base form + affixes
  - **Nouns** - plural form: dog**s**, suffix**es**, baby → bab**ies**
  - **Verbs** - tense: work**ed**, work**ing**, person: work**s**
- Many irregularities... *"women and children begun running away as the wolves showed their teeth"*
- Morphological analysis:
  - input: "am", output: "be" + 1 PERSON + PRESENT
- Lemmatizer: reduce inflectional forms of a word to a common base form
  e.g. children → child, running → run

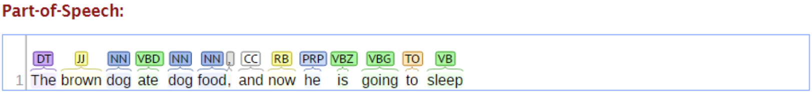
Text Analysis Tasks



Text Analysis Tasks

Part of Speech Tagging

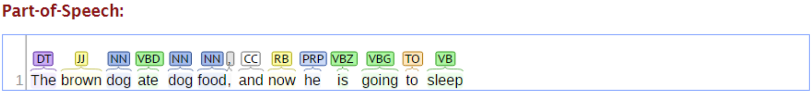
- Tags each word with its part of speech (POS): noun, verb, adjective, adverb, preposition, etc.



Text Analysis Tasks

Part of Speech Tagging

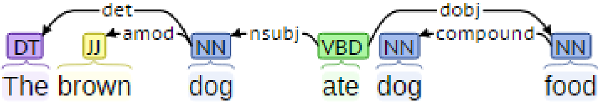
- Tags each word with its part of speech (POS): noun, verb, adjective, adverb, preposition, etc.



- Surrounding words help deciding on the correct POS tag for ambiguous words:
  I'm reading an interesting book $\Rightarrow$ *book* $=$ NOUN
  I would like to book a flight $\Rightarrow$ *book* $=$ VERB

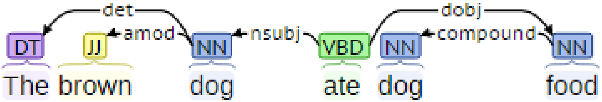
Text Analysis Tasks
Syntactic Parsing

- Analyzes the syntactic structure of a sentence



Text Analysis Tasks
Syntactic Parsing

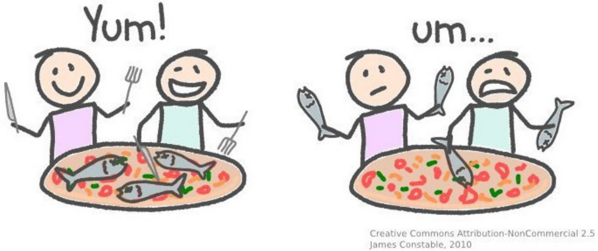
- Analyzes the syntactic structure of a sentence



- Let’s look at some syntactic ambiguities!

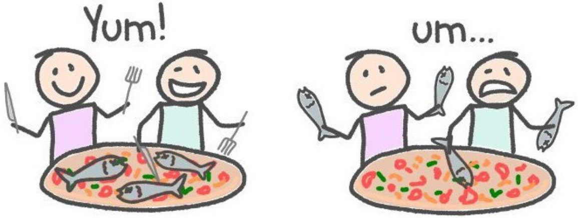
Text Analysis Tasks
Syntactic Parsing

- *"They ate pizza with anchovies"*



Text Analysis Tasks
Syntactic Parsing

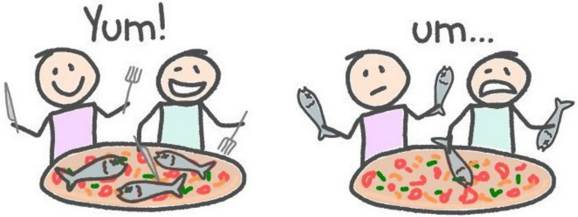
- *"They ate pizza with anchovies"*



- (1) They ate pizza, the pizza had anchovies on it

Text Analysis Tasks
Syntactic Parsing

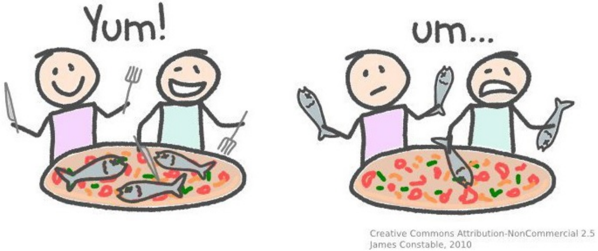
- *"They ate pizza with anchovies"*



- (1) They ate pizza, the pizza had anchovies on it
- (2) They ate pizza using anchovies instead of utensils

Text Analysis Tasks
Syntactic Parsing

- *"They ate pizza with anchovies"*

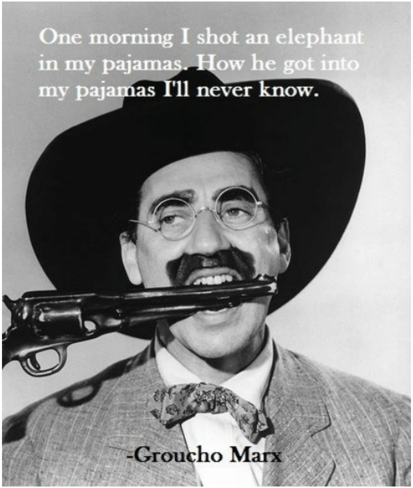


- (1) They ate pizza, the pizza had anchovies on it
- (2) They ate pizza using anchovies instead of utensils
- (3) The anchovies also ate pizza

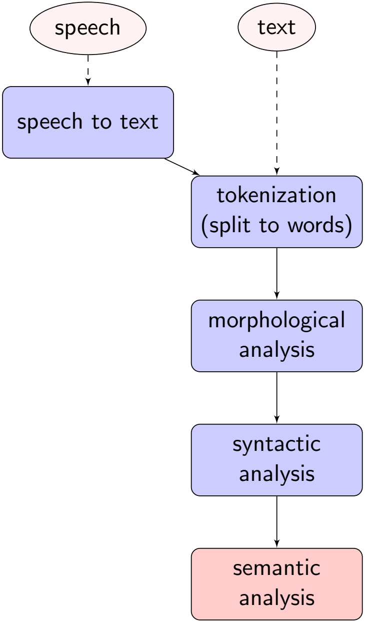
- Each of the interpretations yields a different syntactic analysis

Text Analysis Tasks

Syntactic Parsing

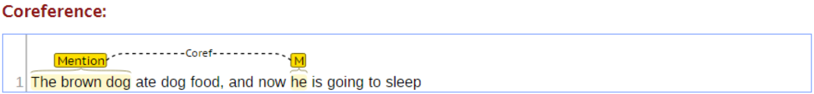


Text Analysis Tasks



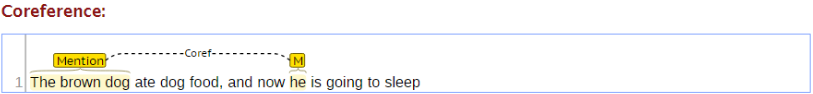
Text Analysis Tasks
Coreference Resolution

- Identify mentions referring to the same entity



Text Analysis Tasks
Coreference Resolution

- Identify mentions referring to the same entity



- Considered a difficult task!

Text Analysis Tasks
Coreference Resolution

- *"I gave the monkeys the bananas because they were* ***hungry****"* $\Rightarrow$
  they $=$ the monkeys

Text Analysis Tasks
Coreference Resolution

- *"I gave the monkeys the bananas because they were* **hungry***"* $\Rightarrow$ they = the monkeys
- *"I gave the monkeys the bananas because they were* **ripe***"* $\Rightarrow$ they = the bananas

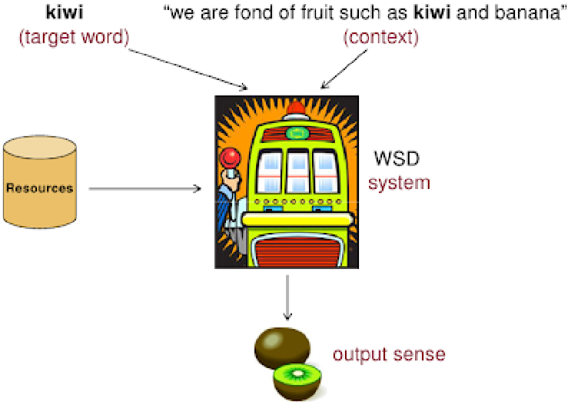
Text Analysis Tasks

Word Sense Disambiguation

- What's the correct sense of a word in a given context?

Text Analysis Tasks
Word Sense Disambiguation

- What's the correct sense of a word in a given context?

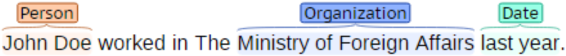


from `http://naviglinlp.blogspot.co.il/`

Text Analysis Tasks

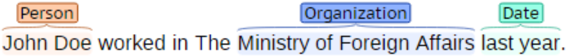
Named Entities

- **Named Entity Recognition:** recognize entities and their type

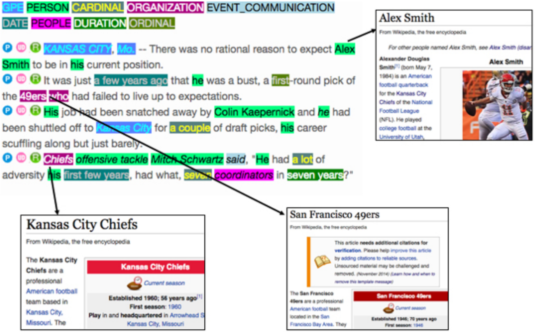


Text Analysis Tasks
Named Entities

- **Named Entity Recognition:** recognize entities and their type



- **Entity Linking:** linking entities to their Wikipedia pages



from `http://www.ibm.com/blogs/research`

NLP is hard!

- Tokenization and POS tagging are almost 100% accurate today, but semantic tasks are far from that

NLP is hard!

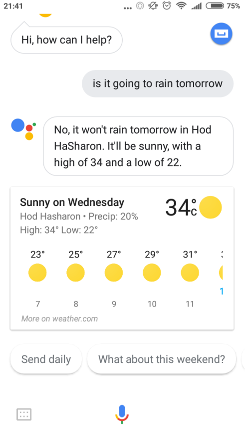
- Tokenization and POS tagging are almost 100% accurate today, but semantic tasks are far from that
- Two major difficulties:

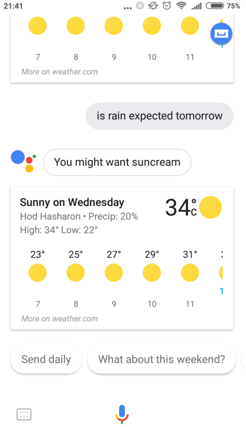
NLP is hard!

- Tokenization and POS tagging are almost 100% accurate today, but semantic tasks are far from that
- Two major difficulties:
  - **Ambiguity**: one text can have multiple meanings

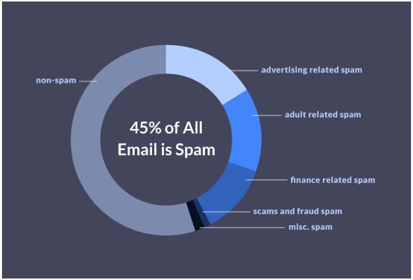
NLP is hard!

- Tokenization and POS tagging are almost 100% accurate today, but semantic tasks are far from that
- Two major difficulties:
  - **Ambiguity**: one text can have multiple meanings
  - **Lexical variability**: the same meaning can be expressed with different words



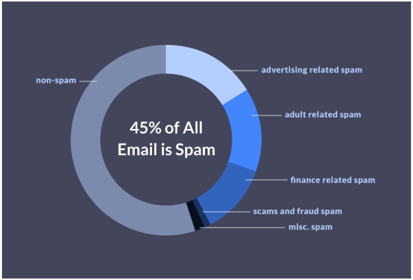


Example Application: Spam Detection



(used to be much worse... $>$ 90%!)

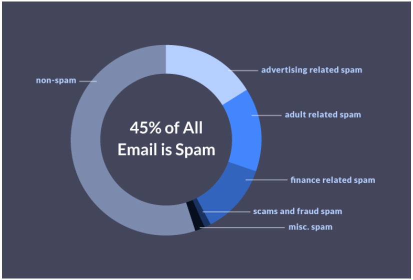
Example Application: Spam Detection



(used to be much worse... $>$ 90%!)

- Automatically determine whether an email is spam or not

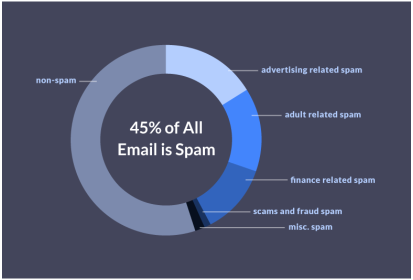
Example Application: Spam Detection



(used to be much worse... $>$ 90%!)

- Automatically determine whether an email is spam or not
  - (and move spam messages to "spam" folder)

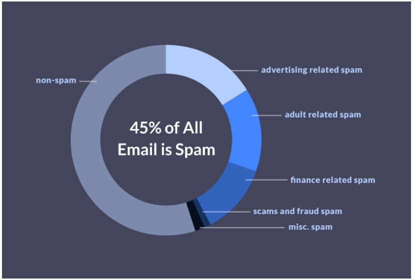
Example Application: Spam Detection



(used to be much worse... $>$ 90%!)

- Automatically determine whether an email is spam or not
  - (and move spam messages to "spam" folder)
- Special case of *Text Classification*: given a text, automatically determine its topic

Example Application: Spam Detection



(used to be much worse... $> 90\%$!)

- Automatically determine whether an email is spam or not
  - (and move spam messages to "spam" folder)
- Special case of *Text Classification*: given a text, automatically determine its topic
- How does it work?

Spam Detection

Let's think of characteristics of spam emails

- Unknown sender

Spam Detection

Let's think of characteristics of spam emails

- Unknown sender
- Spam triggering words:
  - *Earn extra cash*
  - *Earn $*
  - *Free*
  - *Lose weight*
  - *Instant*
  - *Bonus*
  - *...*

Spam Detection

Let's think of characteristics of spam emails

- Unknown sender
- Spam triggering words:
  - *Earn extra cash*
  - *Earn $*
  - *Free*
  - *Lose weight*
  - *Instant*
  - *Bonus*
  - *...*
- Naive idea: mark any email that contains these words as spam

Spam Detection

Let's think of characteristics of spam emails

- Unknown sender
- Spam triggering words:
  - *Earn extra cash*
  - *Earn $*
  - *Free*
  - *Lose weight*
  - *Instant*
  - *Bonus*
  - *...*
- Naive idea: mark any email that contains these words as spam
- **Problem**: inaccurate (will mark non-spam as spam and vice versa)

Spam Detection

Rule-based Approach

- Better idea: define rules, e.g. *"mark as spam if unknown sender and contains at least 2 spam triggering words"*

Spam Detection
Rule-based Approach

- Better idea: define rules, e.g. *"mark as spam if unknown sender and contains at least 2 spam triggering words"*
- More accurate: e.g. will not mark an email from your mother, with the word "instant" as spam :)

Spam Detection
Rule-based Approach

- Better idea: define rules, e.g. *"mark as spam if unknown sender and contains at least 2 spam triggering words"*
- More accurate: e.g. will not mark an email from your mother, with the word "instant" as spam :)
- **Problems**:
  - Finding the optimal rules is difficult

Spam Detection
Rule-based Approach

- Better idea: define rules, e.g. *"mark as spam if unknown sender and contains at least 2 spam triggering words"*
- More accurate: e.g. will not mark an email from your mother, with the word "instant" as spam :)
- **Problems**:
  - Finding the optimal rules is difficult
  - Not all triggering words were created equal

Spam Detection
Rule-based Approach

- Better idea: define rules, e.g. *"mark as spam if unknown sender and contains at least 2 spam triggering words"*
- More accurate: e.g. will not mark an email from your mother, with the word "instant" as spam :)
- **Problems**:
  - Finding the optimal rules is difficult
  - Not all triggering words were created equal
- **Solution**: Let the computer "learn" these rules alone!

Spam Detection
Supervised Learning

I have sent you this message earlier, but your failure to respond has prompted me to re-sending it once again. It is about my late client who lost his life in an automobile accident along with his wife and only child.

I assisted him in making a deposit worth **$10.5M**. The Bank has therefore threatened to seize his account if an heir is not directly specified. You and my late client both share the same last name. With great respect, i want you to stand as an heir to the account so that his deposited funds can be released and transferred to you directly.

Kindly get back to my private email address for more update on this transaction (**richrdbernard65@gmail.com**)

Best Regards

Barrister Richard Bernard.

- Let the computer learn a scoring function:

$$score = \ldots + \alpha_{have} \cdot c(\text{have}) + \alpha_{sent} \cdot c(\text{sent}) + \ldots + \alpha_{bernard} \cdot c(\text{bernard})$$

Spam Detection
Supervised Learning

I have sent you this message earlier, but your failure to respond has prompted me to re-sending it once again. It is about my late client who lost his life in an automobile accident along with his wife and only child.

I assisted him in making a deposit worth **$10.5M**. The Bank has therefore threatened to seize his account if an heir is not directly specified. You and my late client both share the same last name. With great respect, i want you to stand as an heir to the account so that his deposited funds can be released and transferred to you directly.

Kindly get back to my private email address for more update on this transaction (**richrdbernard65@gmail.com**)

Best Regards

Barrister Richard Bernard.

- Let the computer learn a scoring function:
  $score = \ldots + \alpha_{have} \cdot c(\text{have}) + \alpha_{sent} \cdot c(\text{sent}) + \ldots + \alpha_{bernard} \cdot c(\text{bernard})$
- Different weight α_i for each word, e.g. $\alpha_{cash} > \alpha_{document}$

Spam Detection
Supervised Learning

I have sent you this message earlier, but your failure to respond has prompted me to re-sending it once again. It is about my late client who lost his life in an automobile accident along with his wife and only child.

I assisted him in making a deposit worth **$10.5M**. The Bank has therefore threatened to seize his account if an heir is not directly specified. You and my late client both share the same last name. With great respect, i want you to stand as an heir to the account so that his deposited funds can be released and transferred to you directly.

Kindly get back to my private email address for more update on this transaction (**richrdbernard65@gmail.com**)

Best Regards

Barrister Richard Bernard.

- Let the computer learn a scoring function:

  $score = ... + \alpha_{have} \cdot c(\text{have}) + \alpha_{sent} \cdot c(\text{sent}) + ... + \alpha_{bernard} \cdot c(\text{bernard})$
- Different weight α_i for each word, e.g. $\alpha_{cash} > \alpha_{document}$
- Classify as spam if $score > threshold$ (learn threshold too!)

Spam Detection
Supervised Learning

- How does the computer learn the α weights?

Spam Detection
Supervised Learning

- How does the computer learn the α weights?
- **Supervised learning:** estimate a function (learn weights) using labeled examples

Spam Detection
Supervised Learning

- How does the computer learn the α weights?
- **Supervised learning:** estimate a function (learn weights) using labeled examples
- Take a lot of emails, manually mark them as spam/not spam

Spam Detection
Supervised Learning

- How does the computer learn the α weights?
- **Supervised learning:** estimate a function (learn weights) using labeled examples
- Take a lot of emails, manually mark them as spam/not spam
- The computer learns a function (weights) that best predicts spam/not spam for the **known** emails

Spam Detection
Supervised Learning

- How does the computer learn the α weights?
- **Supervised learning:** estimate a function (learn weights) using labeled examples
- Take a lot of emails, manually mark them as spam/not spam
- The computer learns a function (weights) that best predicts spam/not spam for the **known** emails
- If we have enough examples, it would also work well on new emails

Spam Detection
Features

- We used *bag-of-words* as features for classification :
  { I, have, sent, you, ... }

Spam Detection
Features

- We used *bag-of-words* as features for classification : $\{$ I, have, sent, you, ... $\}$
- If we have enough spam examples that contain the word "urgent", α_{urgent} will be high

Spam Detection
Features

- We used *bag-of-words* as features for classification :
  { I, have, sent, you, ... }
- If we have enough spam examples that contain the word "urgent", α_{urgent} will be high
- What about similar words like "immediate" or "instant"?

Spam Detection
Features

- We used *bag-of-words* as features for classification :
  { I, have, sent, you, ... }
- If we have enough spam examples that contain the word "urgent", α_{urgent} will be high
- What about similar words like "immediate" or "instant"?
- We need to find a way to let the computer know about semantically-similar words

Word Representation
One-hot Vectors

- How do we represent all the words in the computer?

Word Representation

One-hot Vectors

- How do we represent all the words in the computer?
- **Simplest:** we have a dictionary, and each word has an index, e.g. $\textit{index}(\text{urgent}) = 316$, $\textit{index}(\text{instant}) = 12418$

Word Representation

One-hot Vectors

- How do we represent all the words in the computer?
- **Simplest:** we have a dictionary, and each word has an index, e.g. $index(\text{urgent}) = 316$, $index(\text{instant}) = 12418$
- You can think of the word with index i as a vector (array of numbers) with zeros and one entry with 1 in the ith index - *"one-hot vector"*:

urgent	0	0	...	1	0	...	0	0	...	0

$\uparrow$ 316

instant	0	0	...	0	0	...	1	0	...	0

$\uparrow$ 12418

Spam Detection
Bag-of-words with One-hot Vectors

- A vector representing the entire email: sum of one-hot vectors of the words in the email:

I	0	0	...	1	0	...	0	0	...	0
have	0	1	...	0	0	...	0	0	...	0
sent	0	0	...	0	0	...	1	0	...	0
+ ...						...				
bernard	0	0	...	0	1	...	0	0	...	0
=										
feature vector	0	4	...	2	1	...	1	0	...	0

Spam Detection
Bag-of-words with One-hot Vectors

- A vector representing the entire email: sum of one-hot vectors of the words in the email:

I	0	0	...	1	0	...	0	0	...	0
have	0	1	...	0	0	...	0	0	...	0
sent	0	0	...	0	0	...	1	0	...	0
+ ...						...				
bernard	0	0	...	0	1	...	0	0	...	0
=										
feature vector	0	4	...	2	1	...	1	0	...	0

- **Problem**: Emails with similar words (e.g. *deliver* instead of *send*, *urgent* instead of *instant*) have very different feature vectors!

Word Representation
Distributional Word Vectors

- Can we have similar vectors for semantically-similar words?

Word Representation
Distributional Word Vectors

- Can we have similar vectors for semantically-similar words?
- "*You shall know a word by the company it keeps*"
  (John Rupert Firth, 1957)

Word Representation

Distributional Word Vectors

- Can we have similar vectors for semantically-similar words?
- "*You shall know a word by the company it keeps*"
  (John Rupert Firth, 1957)

elevator	0	0	...	0.16	0	...	0.49	0	...	0
lift	0	0	...	0.15	0	...	0.51	0	...	0
				↑			↑			
				up			stairs			

Word Representation

Distributional Word Vectors

- Can we have similar vectors for semantically-similar words?
- "*You shall know a word by the company it keeps*"
  (John Rupert Firth, 1957)

elevator	0	0	...	0.16	0	...	0.49	0	...	0
lift	0	0	...	0.15	0	...	0.51	0	...	0
				up			stairs			

- Now semantically-similar words have similar word vectors!

Spam Detection

Bag-of-words with Distributional Word Vectors

- Again, we sum up all the vectors:

I	0	0	...	0.12	0.03	...	...	...	...	0.04	0	...	0
have	0	0.22	...	0	0	...	...	...	...	0	0	...	0
sent	0	0.43	...	0	0.1	...	...	...	...	0.25	0	...	0
+ ...						...							
bernard	0	0	...	0	0.67	...	...	...	...	0	0	...	0
=													
FV	0	0.65	...	0.12	0.71	...	...	...	...	0.29	0	...	0

Spam Detection

Bag-of-words with Distributional Word Vectors

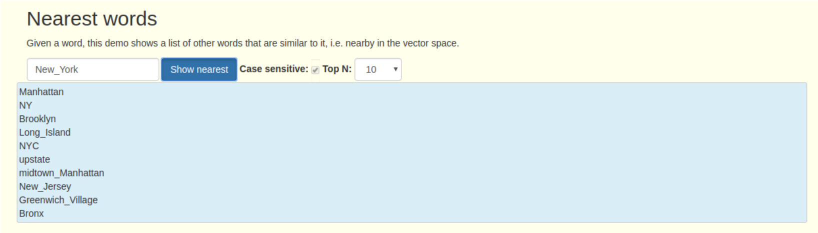
- Again, we sum up all the vectors:

I	0	0	...	0.12	0.03	...	...	...	...	0.04	0	...	0
have	0	0.22	...	0	0	...	...	...	...	0	0	...	0
sent	0	0.43	...	0	0.1	...	...	...	...	0.25	0	...	0
+ ...						...							
bernard	0	0	...	0	0.67	...	...	...	...	0	0	...	0
=													
FV	0	0.65	...	0.12	0.71	...	...	...	...	0.29	0	...	0

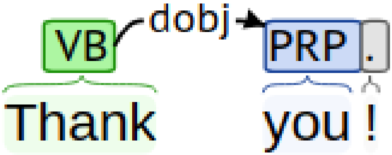
- We can now replace a word (e.g. *sent*) with a similar word (e.g. *delivered*) and get a similar feature vector $\Rightarrow$ same classification for similar emails!

Word Embeddings

- [A more recent type of distributional vectors]
- Find most similar words:



See more here: `http://bionlp-www.utu.fi/wv_demo/`


VB
dobj
PRP
.
Thank
you
!

Additional Resources

- Books:
  - Chris Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999.
  - Dan Jurafsky and James H. Martin, Speech and Language Processing. Second Edition. Pearson Education, 2014.
- Resources from NACLO - North American Computational Linguistics Olympiad
  `http://nacloweb.org/resources.php`
- My blog: `http://veredshwartz.blogspot.co.il`