Natural Language Processing

Lexical semantics

Yulia Tsvetkov

yuliats@cs.washington.edu
Announcements

- HW1 submission this week
- This week’s quiz is on Wednesday
  - LR, LMs
- Yulia - no OHs this week
  - Please use TAs’ OHs if needed
Language models recap
The Language Modeling problem

- Assign a probability to every sentence (or any string of words)
  - finite vocabulary (e.g. words or characters)
  - infinite set of sequences

\[
\sum_{e \in \Sigma^*} p_{LM}(e) = 1
\]

\[
p_{LM}(e) \geq 0 \quad \forall e \in \Sigma^*
\]
Language Modeling

- If we have some text, then the probability of this text (according to the Language Model) is:

\[
P(x^{(1)}, \ldots, x^{(T)}) = P(x^{(1)}) \times P(x^{(2)} | x^{(1)}) \times \cdots \times P(x^{(T)} | x^{(T-1)}, \ldots, x^{(1)})
\]

\[
= \prod_{t=1}^{T} P(x^{(t)} | x^{(t-1)}, \ldots, x^{(1)})
\]

This is what our LM provides
n-gram Language Models

“I have a dog whose name is Lucy. I have two cats, they like playing with Lucy.”

- **Definition:** An n-gram is a chunk of n consecutive words.
  - unigrams: {I, have, a, dog, whose, name, is, Lucy, two, cats, they, like, playing, with}
  - bigrams: {I have, have a, a dog, dog whose, … , with Lucy}
  - trigrams: {I have a, have a dog, a dog whose, … , playing with Lucy}
  - four-grams: {I have a dog, … , like playing with Lucy}
  - …
“I have a dog whose name is Lucy. I have two cats, they like playing with Lucy.”

- corpus size $m = 17$
- $P(\text{Lucy}) = 2/17$; $P(\text{cats}) = 1/17$

Unigram probability: $P(w) = \frac{\text{count}(w)}{m} = \frac{C(w)}{m}$
bigram probability

“I have a dog whose name is Lucy. I have two cats, they like playing with Lucy.”

\[
P(A | B) = \frac{P(A, B)}{P(B)}
\]

\[
P(\text{have} | \text{I}) = \frac{P(\text{I have})}{P(\text{I})} = \frac{2}{2} = 1
\]

\[
P(\text{two} | \text{have}) = \frac{P(\text{have two})}{P(\text{have})} = \frac{1}{2} = 0.5
\]

\[
P(\text{eating} | \text{have}) = \frac{P(\text{have eating})}{P(\text{have})} = \frac{0}{2} = 0
\]

\[
P(w_2 | w_1) = \frac{C(w_1, w_2)}{\sum_w C(w_1, w)} = \frac{C(w_1, w_2)}{C(w_1)}
\]
trigram probability

“I have a dog whose name is Lucy. I have two cats, they like playing with Lucy.”

\[
P(A | B) = \frac{P(A, B)}{P(B)}
\]

\[
P(a | I \text{ have}) = \frac{C(I \text{ have } a)}{C(I \text{ have})} = \frac{1}{2} = 0.5
\]

\[
P(\text{several} | I \text{ have}) = \frac{C(I \text{ have } \text{several})}{C(I \text{ have})} = \frac{0}{2} = 0
\]

\[
P(w_3 | w_1 w_2) = \frac{C(w_1, w_2, w_3)}{\sum_w C(w_1, w_2, w)} = \frac{C(w_1, w_2, w_3)}{C(w_1, w_2)}
\]
n-gram probability

“I have a dog whose name is Lucy. I have two cats, they like playing with Lucy.”

\[
P(A | B) = \frac{P(A,B)}{P(B)}
\]

\[
P(w_i | w_1, w_2, ..., w_{i-1}) = \frac{C(w_i, w_2, ..., w_{i-1}, w_i)}{C(w_1, w_2, ..., w_{i-1})}
\]
Markov assumption

- We make the Markov assumption: $x^{(t+1)}$ depends only on the preceding $n-1$ words
  - Markov chain is a “…stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.”

$$P(x^{(t+1)}|x^{(t)}, \ldots, x^{(1)}) = P(x^{(t+1)}|x^{(t)}, \ldots, x^{(t-n+2)})$$

$n-1$ words
Calculating a probability of a sequence

Chain rule

\[
p(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \\
p(X_1 = x_1) \prod_{i=2}^{n} p(X_i = x_i \mid X_1 = x_1, \ldots, X_{i-1} = x_{i-1})
\]
Second-order Markov process:

- Using independence assumption:

\[
p(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) =
\]
\[
p(X_1 = x_1) \times p(X_2 = x_2 | X_1 = x_1)
\]
\[
\times \prod_{i=3}^{n} p(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})
\]
Example

\[ p(\text{the dog barks STOP}) = q(\text{the} \mid *, *) \times q(\text{dog} \mid *, \text{the}) \times q(\text{barks} \mid \text{the}, \text{dog}) \times q(\text{STOP} \mid \text{dog, barks}) \times \]
Intuitively, language models should assign high probability to real language they have not seen before

- Want to maximize likelihood on held-out, not training data
- Models derived from counts / sufficient statistics require generalization parameters to be tuned on held-out data to simulate test generalization
- Set hyperparameters to maximize the likelihood of the held-out data (usually with grid search or EM)

**Evaluation**

- Build language model from a train set
- Tune the model's parameters on a validation set
- Evaluate the model on a test set

Counts / parameters from here

Hyperparameters from here

Evaluate here
Evaluation

- **Extrinsic** evaluation: build a new language model, use it for some task (MT, ASR, etc.)
- **Intrinsic**: measure how good we are at modeling language
Extrinsic evaluation of N-gram models

- Best evaluation for comparing models A and B
  - Put each model in a task
    - spelling corrector, speech recognizer, MT system
  - Run the task, get an accuracy for A and for B
    - How many misspelled words corrected properly
    - How many words translated correctly

- Compare accuracy for A and B
Difficulty of extrinsic (in-vivo) evaluation of N-gram models

- Extrinsic evaluation
  - Time-consuming; can take days or weeks

So

- Sometimes use intrinsic evaluation: **perplexity**
  - Bad approximation
    - unless the test data looks just like the training data
  - So generally only useful in pilot experiments
  - But is helpful to think about
Intrinsic evaluation: perplexity

- Test data: $S = \{s_1, s_2, \ldots, s_{sent}\}$
  - parameters are estimated on training data

$$p(S) = \prod_{i=1}^{sent} p(s_i)$$

- $sent$ is the number of sentences in the test data
Evaluation: perplexity

- Test data: $S = \{s_1, s_2, \ldots, s_{sent}\}$
  - parameters are estimated on training data

$$p(S) = \prod_{i=1}^{sent} p(s_i)$$

- $sent$ is the number of sentences in the test data

\[
p(\text{the dog barks STOP}) = q(\text{the} | *, *) \times \\
q(\text{dog} | *, \text{the}) \times \\
q(\text{barks} | \text{the, dog}) \times \\
q(\text{STOP} | \text{dog, barks})
\]
Evaluation: perplexity

- Test data: $S = \{s_1, s_2, \ldots, s_{sent}\}$
  - parameters are estimated on training data

\[
p(S) = \prod_{i=1}^{sent} p(s_i)
\]

\[
\log_2 p(S) = \sum_{i=1}^{sent} \log_2 p(s_i)
\]

- $sent$ is the number of sentences in the test data
Evaluation: perplexity

● Test data: \( S = \{s_1, s_2, \ldots, s_{\text{sent}}\} \)
  ○ parameters are estimated on training data

\[
p(S) = \prod_{i=1}^{\text{sent}} p(s_i)
\]

\[
\log_2 p(S) = \sum_{i=1}^{\text{sent}} \log_2 p(s_i)
\]

perplexity = \( 2^{-l} \), \( l = \frac{1}{M} \sum_{i=1}^{\text{sent}} \log_2 p(s_i) \)

○ \text{sent} is the number of sentences in the test data
○ \( M \) is the number of words in the test corpus
Evaluation: perplexity

- Test data: $S = \{s_1, s_2, \ldots, s_{\text{sent}}\}$
  - parameters are estimated on training data

\[ p(S) = \prod_{i=1}^{\text{sent}} p(s_i) \]
\[ \log_2 p(S) = \sum_{i=1}^{\text{sent}} \log_2 p(s_i) \]
\[ \text{perplexity} = 2^{-l}, \quad l = \frac{1}{M} \sum_{i=1}^{\text{sent}} \log_2 p(s_i) \]

- $\text{sent}$ is the number of sentences in the test data
- $M$ is the number of words in the test corpus
- A good language model has high $p(S)$ and low perplexity
Understanding perplexity

\[
\text{perplexity} = 2^{- \frac{1}{M} \sum_{i=1}^{\text{sent}} \log_2 p(s_i)}
\]

- It’s a branching factor
  - assign probability of 1 to the test data \( \Rightarrow \) perplexity = 1
  - assign probability of \( 1/|V| \) to every word \( \Rightarrow \) perplexity = \( |V| \)
  - assign probability of 0 to anything \( \Rightarrow \) perplexity = \( \infty \)
    - this motivates the proper probability constraint
      \[
      \sum_{e \in \Sigma^*} p_{LM}(e) = 1
      \]
      \[
      p_{LM}(e) \geq 0 \quad \forall e \in \Sigma^*
      \]

- cannot compare perplexities of LMs trained on different corpora
Lexical semantics
What do words mean?

- N-gram or text classification methods we've seen so far
  - Words are just strings (or indices $w_i$ in a vocabulary list)
  - That's not very satisfactory!
What are various ways to represent the meaning of a word?
Desiderata

What should a theory of word meaning do for us?
Let's look at some desiderata from *lexical semantics*, the linguistic study of word meaning.
Lexical semantics

- How should we represent the meaning of the word?
  - Words, lemmas, senses, definitions

- The OED defines the word 'pepper' as:
  - "A hot pungent spice derived from the prepared fruits (peppercorn) of the pepper plant, Piper nigrum (one sense is used from early times to season food, either whole or ground to powder (often in association with salt). Also (locally, chiefly with distinguishing word): a similar spice derived from the fruits of certain other species of the genus Piper; the fruits themselves."

- A plant, Piper nigrum (family Piperaceae), a climbing shrub indigenous to South Asia, also cultivated elsewhere in the tropics, which has alternate, toothed, entire leaves, with pendulous spikes of small green flowers opposite the leaves, succeeded by small berries turning red when ripe. Also more widely: any plant of the genus Piper or the family Piperaceae.

- "A. With distinguishing word: any of numerous plants of other families having hot pungent fruits or leaves which resemble pepper (1a) in taste and in some cases are used as a substitute for it."

- For more detailed information, visit the OED at http://www.oed.com/
Lemmas and senses

A sense or “concept” is the meaning component of a word. Lemmas can be polysemous (have multiple senses).

Word: mouse

1. any of numerous small rodents...
2. a hand-operated device that controls a cursor...

Modified from the online thesaurus WordNet
Relation: synonymity

- Synonyms have the same meaning in some or all contexts.
  - filbert / hazelnut
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - Water / H$_2$O
The Linguistic Principle of Contrast

Difference in form $\rightarrow$ difference in meaning

- Note that there are probably no examples of perfect synonymy
  - Even if many aspects of meaning are identical
  - Still may not preserve the acceptability based on notions of politeness, slang, register, genre, etc.
    - Water / H20 in a surfing guide?
    - my big sister $\neq$ my large sister
Relation: antonymy

Senses that are opposites with respect to one feature of meaning

- Otherwise, they are very similar!
  - dark/light  short/long  fast/slow  rise/fall
  - hot/cold  up/down  in/out

More formally: antonyms can

- define a binary opposition or be at opposite ends of a scale
  - long/short, fast/slow

- be reversives:
  - rise/fall, up/down
Relation: similarity

Words with similar meanings.

- Not synonyms, but sharing some element of meaning
  - car, bicycle
  - cow, horse
Ask humans how similar two words are

<table>
<thead>
<tr>
<th>word1</th>
<th>word2</th>
<th>similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>vanish</td>
<td>disappear</td>
<td>9.8</td>
</tr>
<tr>
<td>behave</td>
<td>obey</td>
<td>7.3</td>
</tr>
<tr>
<td>belief</td>
<td>impression</td>
<td>5.95</td>
</tr>
<tr>
<td>muscle</td>
<td>bone</td>
<td>3.65</td>
</tr>
<tr>
<td>modest</td>
<td>flexible</td>
<td>0.98</td>
</tr>
<tr>
<td>hole</td>
<td>agreement</td>
<td>0.3</td>
</tr>
</tbody>
</table>

SimLex-999 dataset (Hill et al., 2015)
Relation: word relatedness

Also called "word association"

- Words be related in any way, perhaps via a semantic frame or field
  - car, bicycle: similar
  - car, gasoline: related, not similar
Semantic field

Words that

- cover a particular semantic domain
- bear structured relations with each other

hospitals

- surgeon, scalpel, nurse, anaesthetic, hospital

restaurants

- waiter, menu, plate, food, menu, chef,

houses

- door, roof, kitchen, family, bed
Relation: superordinate/ subordinate

● One sense is a subordinate (hyponym) of another if the first sense is more specific, denoting a subclass of the other
  ○ car is a subordinate of vehicle
  ○ mango is a subordinate of fruit

● Conversely superordinate (hypernym)
  ○ vehicle is a superordinate of car
  ○ fruit is a superordinate of mango
Taxonomy

![Taxonomy Diagram]

- **Superordinate**
  - chair
  - furniture

- **Basic**
  - office chair
  - piano chair
  - rocking chair
  - lamp
  - torchiere
  - desk lamp
  - table
  - end table
  - coffee table

- **Subordinate**
Lexical semantics

- How should we represent the meaning of the word?
  - Dictionary definition
  - Lemma and wordforms
  - Senses
  - Relationships between words or senses
  - Taxonomic relationships
  - Word similarity, word relatedness
  - Semantic frames and roles
  - Connotation and sentiment
Lexical semantics

- How should we represent the meaning of the word?
  - Dictionary definition
  - Lemma and wordforms
  - Senses
  - Relationships between words or senses
  - Taxonomic relationships
  - Word similarity, word relatedness
  - Semantic frames and roles
    - *John hit Bill*
    - *Bill was hit by John*
Lexical Semantics

- How should we represent the meaning of the word?
  - Dictionary definition
  - Lemma and wordforms
  - Senses
  - Relationships between words or senses
  - Taxonomic relationships
  - Word similarity, word relatedness
  - Semantic frames and roles
  - Connotation and sentiment
    - **valence**: the pleasantness of the stimulus
    - **arousal**: the intensity of emotion
    - **dominance**: the degree of control exerted by the stimulus

<table>
<thead>
<tr>
<th></th>
<th>Valence</th>
<th>Arousal</th>
<th>Dominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>courageous</td>
<td>8.05</td>
<td>5.5</td>
<td>7.38</td>
</tr>
<tr>
<td>music</td>
<td>7.67</td>
<td>5.57</td>
<td>6.5</td>
</tr>
<tr>
<td>heartbeat</td>
<td>2.45</td>
<td>5.65</td>
<td>3.58</td>
</tr>
<tr>
<td>cub</td>
<td>6.71</td>
<td>3.95</td>
<td>4.24</td>
</tr>
<tr>
<td>life</td>
<td>6.68</td>
<td>5.59</td>
<td>5.89</td>
</tr>
</tbody>
</table>
Electronic Dictionaries

WordNet

https://wordnet.princeton.edu/
Electronic Dictionaries

WordNet

```python
from nltk.corpus import wordnet as wn
panda = wn.synset('panda.n.01')
hyper = lambda s: s.hypernyms()
list(panda.closure(hyper))
```

[Synset('procyonid.n.01'),
 Synset('carnivore.n.01'),
 Synset('placental.n.01'),
 Synset('mammal.n.01'),
 Synset('vertebrate.n.01'),
 Synset('chordate.n.01'),
 Synset('animal.n.01'),
 Synset('organism.n.01'),
 Synset('living_thing.n.01'),
 Synset('whole.n.02'),
 Synset('object.n.01'),
 Synset('physical_entity.n.01'),
 Synset('entity.n.01')]
Problems with discrete representations

- Too coarse
  - expert ↔ skillful
- Sparse
  - wicked, badass, ninja
- Subjective
- Expensive
- Hard to compute word relationships

```
expert  [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
skillful [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
```

- dimensionality: PTB: 50K, Google1T 13M
Distributional hypothesis

“The meaning of a word is its use in the language”

[Wittgenstein PI 43]

“You shall know a word by the company it keeps”

[Firth 1957]

If A and B have almost identical environments we say that they are synonyms.

[Harris 1954]
Example

What does ongchoi mean?
Example

- Suppose you see these sentences:
  - Ongchoi is delicious sautéed with garlic.
  - Ongchoi is superb over rice
  - Ongchoi leaves with salty sauces

- And you've also seen these:
  - …spinach sautéed with garlic over rice
  - Chard stems and leaves are delicious
  - Collard greens and other salty leafy greens
Ongchoi: Ipomoea aquatica "Water Spinach"

Ongchoi is a leafy green like spinach, chard, or collard greens
Model of meaning focusing on similarity

- Each word = a vector
  - not just “word” or word45.
  - similar words are “nearby in space”
  - We build this space automatically by seeing which words are nearby in text
We define meaning of a word as a vector

- Called an "embedding" because it's embedded into a space
- The standard way to represent meaning in NLP

Every modern NLP algorithm uses embeddings as the representation of word meaning
Intuition: why vectors?

Consider sentiment analysis:

- **With words**, a feature is a word identity
  - Feature 5: 'The previous word was "terrible"'
  - requires **exact same word** to be in training and test

- **With embeddings:**
  - Feature is a word vector
  - 'The previous word was vector [35,22,17…]
  - Now in the test set we might see a similar vector [34,21,14]
  - We can generalize to **similar but unseen** words!!!
There are many kinds of embeddings

- **Count-based**
  - Words are represented by a simple function of the counts of nearby words

- **Class-based**
  - Representation is created through hierarchical clustering, Brown clusters

- **Distributed prediction-based (type) embeddings**
  - Representation is created by training a classifier to distinguish nearby and far-away words: word2vec, fasttext

- **Distributed contextual (token) embeddings from language models**
  - ELMo, BERT
We'll discuss 2 kinds of embeddings

- **tf-idf**
  - Information Retrieval workhorse!
  - A common baseline model
  - Sparse vectors
  - Words are represented by (a simple function of) the counts of nearby words

- **Word2vec**
  - Dense vectors
  - Representation is created by training a classifier to predict whether a word is likely to appear nearby
  - Later we'll discuss extensions called contextual embeddings
Vector Semantics
## Term-document matrix

<table>
<thead>
<tr>
<th></th>
<th>As You Like It</th>
<th>Twelfth Night</th>
<th>Julius Caesar</th>
<th>Henry V</th>
</tr>
</thead>
<tbody>
<tr>
<td>battle</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>soldier</td>
<td>2</td>
<td>80</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>fool</td>
<td>36</td>
<td>58</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>clown</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Context = appearing in the same document.
Term-document Matrix

<table>
<thead>
<tr>
<th></th>
<th>As You Like It</th>
<th>Twelfth Night</th>
<th>Julius Caesar</th>
<th>Henry V</th>
</tr>
</thead>
<tbody>
<tr>
<td>battle</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>soldier</td>
<td>2</td>
<td>80</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>fool</td>
<td>36</td>
<td>58</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>clown</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Each document is represented by a vector of words
Vectors are the basis of information retrieval

- Vectors are similar for the two comedies
- Different than the history
- Comedies have more fools and wit and fewer battles.

<table>
<thead>
<tr>
<th></th>
<th>As You Like It</th>
<th>Twelfth Night</th>
<th>Julius Caesar</th>
<th>Henry V</th>
</tr>
</thead>
<tbody>
<tr>
<td>battle</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>soldier</td>
<td>2</td>
<td>80</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>fool</td>
<td>36</td>
<td>58</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>clown</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Visualizing Document Vectors

Example of document vectors plotted in a 2D space with axes labeled as "foul" and "battle". The positions of "Henry V", "Julius Caesar", "As You Like It", and "Twelfth Night" are shown with their corresponding coordinates [4,13], [1,7], [36,1], and [58,0] respectively.
Words can be vectors too

<table>
<thead>
<tr>
<th></th>
<th>As You Like It</th>
<th>Twelfth Night</th>
<th>Julius Caesar</th>
<th>Henry V</th>
</tr>
</thead>
<tbody>
<tr>
<td>battle</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>good</td>
<td>114</td>
<td>80</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>fool</td>
<td>36</td>
<td>58</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>clown</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

- battle is "the kind of word that occurs in Julius Caesar and Henry V"
- fool is "the kind of word that occurs in comedies, especially Twelfth Night"
More common: word-word matrix ("term-context matrix")

<table>
<thead>
<tr>
<th></th>
<th>knife</th>
<th>dog</th>
<th>sword</th>
<th>love</th>
<th>like</th>
</tr>
</thead>
<tbody>
<tr>
<td>knife</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>dog</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>sword</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>love</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>like</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

- Two words are “similar” in meaning if their context vectors are similar
  - Similarity == relatedness
Term-context matrix

Two **words** are similar in meaning if their context vectors are similar

- is traditionally followed by **cherry**
- often mixed, such as **strawberry**
- computer peripherals and personal
- a computer. This includes **digital**
- **information**
- pie, a traditional dessert
- rhubarb pie. Apple pie
- assistants. These devices usually
- available on the internet

<table>
<thead>
<tr>
<th></th>
<th>aardvark</th>
<th>computer</th>
<th>data</th>
<th>result</th>
<th>pie</th>
<th>sugar</th>
</tr>
</thead>
<tbody>
<tr>
<td>cherry</td>
<td>0</td>
<td>...</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>442</td>
</tr>
<tr>
<td>strawberry</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td>digital</td>
<td>0</td>
<td>...</td>
<td>1670</td>
<td>1683</td>
<td>85</td>
<td>5</td>
</tr>
<tr>
<td>information</td>
<td>0</td>
<td>...</td>
<td>3325</td>
<td>3982</td>
<td>378</td>
<td>5</td>
</tr>
</tbody>
</table>
Cosine for computing word similarity

The dot product between two vectors is a scalar:

$$\text{dot product}(v, w) = v \cdot w = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

- The dot product tends to be high when the two vectors have large values in the same dimensions
- Dot product can thus be a useful similarity metric between vectors
Problem with raw dot-product

- Dot product favors long vectors
  - Dot product is higher if a vector is longer (has higher values in many dimension)

Vector length:

$$|v| = \sqrt{\sum_{i=1}^{N} v_i^2}$$

- Frequent words (of, the, you) have long vectors (since they occur many times with other words).
  - So dot product overly favors frequent words
Alternative: cosine for computing word similarity

Based on the definition of the dot product between two vectors $a$ and $b$

$$
a \cdot b = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}
$$

$$
a \cdot b = |a||b| \cos \theta
$$

$$
\frac{a \cdot b}{|a||b|} = \cos \theta
$$
Cosine as a similarity metric

-1: vectors point in opposite directions
+1: vectors point in same directions
0: vectors are orthogonal

- But since raw frequency values are non-negative, the cosine for term-term matrix vectors ranges from 0–1
Cosine examples

\[
\cos(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\|\|\vec{w}\|} = \frac{\vec{v} \cdot \vec{w}}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}
\]

<table>
<thead>
<tr>
<th></th>
<th>pie</th>
<th>data</th>
<th>computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>cherry</td>
<td>442</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>digital</td>
<td>114</td>
<td>80</td>
<td>62</td>
</tr>
<tr>
<td>information</td>
<td>36</td>
<td>58</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\cos(\text{cherry, information}) = \frac{442 \times 5 + 8 \times 3982 + 2 \times 3325}{\sqrt{442^2 + 8^2 + 2^2} \sqrt{5^2 + 3982^2 + 3325^2}} = .017
\]

\[
\cos(\text{digital, information}) = \frac{5 \times 5 + 1683 \times 3982 + 1670 \times 3325}{\sqrt{5^2 + 1683^2 + 1670^2} \sqrt{5^2 + 3982^2 + 3325^2}} = .996
\]
Visualizing angles

**Dimension 1:** ‘pie’

**Dimension 2:** ‘computer’
Count-based representations

<table>
<thead>
<tr>
<th></th>
<th>As You Like It</th>
<th>Twelfth Night</th>
<th>Julius Caesar</th>
<th>Henry V</th>
</tr>
</thead>
<tbody>
<tr>
<td>battle</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>good</td>
<td>114</td>
<td>80</td>
<td>62</td>
<td>89</td>
</tr>
<tr>
<td>fool</td>
<td>36</td>
<td>58</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>wit</td>
<td>20</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

- Counts: term-frequency
  - remove stop words
  - use $\log_{10}(tf)$
  - normalize by document length
But raw frequency is a bad representation

- The co-occurrence matrices we have seen represent each cell by word frequencies
- Frequency is clearly useful; if *sugar* appears a lot near *apricot*, that's useful information
- But overly frequent words like *the*, *it*, or *they* are not very informative about the context
- It's a paradox! How can we balance these two conflicting constraints?
Two common solutions for word weighting

**tf-idf**: tf-idf value for word \( t \) in document \( d \):

\[
w_{t,d} = tf_{t,d} \times idf_t
\]

Words like "the" or "it" have very low idf.

**PMI**: Pointwise mutual information

\[
PMI(w_1, w_2) = \log \frac{p(w_1, w_2)}{p(w_1)p(w_2)}
\]

See if words like "good" appear more often with "great" than we would expect by chance.
TF-IDF

- What to do with words that are evenly distributed across many documents?

\[
\text{tf}_{t,d} = \log_{10}(\text{count}(t,d) + 1)
\]

\[
\text{idf}_i = \log \left( \frac{N}{\text{df}_i} \right)
\]

Words like "the" or "good" have very low idf

\[
\text{wt,d} = \text{tf}_{t,d} \times \text{idf}_t
\]
Positive Pointwise Mutual Information (PPMI)

- In word-context matrix
- Do words $w$ and $c$ co-occur more than if they were independent?

$$\text{PMI}(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)}$$

$$\text{PPMI}(w, c) = \max(\log_2 \frac{P(w, c)}{P(w)P(c)}, 0)$$

- PMI is biased toward infrequent events
  - Very rare words have very high PMI values
  - Give rare words slightly higher probabilities $\alpha=0.75$

$$\text{PPMI}_\alpha(w, c) = \max(\log_2 \frac{P(w, c)}{P(w)P_\alpha(c)}, 0)$$

$$P_\alpha(c) = \frac{\text{count}(c)^\alpha}{\sum_c \text{count}(c)^\alpha}$$
<table>
<thead>
<tr>
<th># name</th>
<th>formula</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Joint probability</td>
<td>$p(x</td>
<td>y)$</td>
</tr>
<tr>
<td>2. Conditional probability</td>
<td>$p(y</td>
<td>x)$</td>
</tr>
<tr>
<td>3. Reverse cond. probability</td>
<td>$p(x</td>
<td>y)$</td>
</tr>
<tr>
<td>4. Pointwise mutual inf. (MI)</td>
<td>$\log \frac{p(x</td>
<td>y)}{p(x)p(y)}$</td>
</tr>
<tr>
<td>5. Mutual dependency (MD)</td>
<td>$\log \frac{p(x</td>
<td>y)}{p(x)p(y)}$</td>
</tr>
<tr>
<td>6. Log frequency biased MD</td>
<td>$\log \frac{p(x</td>
<td>y)}{f(x</td>
</tr>
<tr>
<td>7. Normalized expectation</td>
<td>$\frac{f(x</td>
<td>y)}{f(x</td>
</tr>
<tr>
<td>8. Mutual expectation</td>
<td>$\log p(x</td>
<td>y) + \log p(y)$</td>
</tr>
<tr>
<td>9. Salience</td>
<td>$\log \frac{p(x</td>
<td>y)}{p(x)p(y)}$</td>
</tr>
<tr>
<td>10. Pearson's $\chi^2$ test</td>
<td>$\frac{\sum (x</td>
<td>y)-f(x</td>
</tr>
<tr>
<td>11. Fisher's exact test</td>
<td>$\frac{N!}{x!(N-x)!} \cdot \frac{y!(N-y)!}{(N-N)!}$</td>
<td>(Pedersen, 1996)</td>
</tr>
<tr>
<td>12. t test</td>
<td>$\frac{\bar{x} \pm t_{x,y}^{(1-\alpha/2)} \cdot S x/y}{\sqrt{n}}$</td>
<td>(Church and Hanks, 1990)</td>
</tr>
<tr>
<td>13. z score</td>
<td>$\frac{\bar{y} \pm z_{x,y}^{(1-\alpha/2)} \cdot s x/y}{\sqrt{n}}$</td>
<td>(Berry-Rogge, 1973)</td>
</tr>
<tr>
<td>14. Poisson significance</td>
<td>$\frac{\sum f(x</td>
<td>y)}{\log N}$</td>
</tr>
<tr>
<td>15. Log likelihood ratio</td>
<td>$-2 \sum \log \frac{f(x</td>
<td>y)}{f(x</td>
</tr>
<tr>
<td>16. Squared log likelihood ratio</td>
<td>$-2 \sum \log \frac{f(x</td>
<td>y)}{f(x</td>
</tr>
<tr>
<td>17. Russel-Rao</td>
<td>$\frac{a+b}{a+b+c+d}$</td>
<td>(Russel and Rao, 1940)</td>
</tr>
<tr>
<td>18. Sokal-Michener</td>
<td>$\frac{a+b+c}{a+b+c}$</td>
<td>(Sokal and Michener, 1958)</td>
</tr>
<tr>
<td>19. Rogers-Tanimoto</td>
<td>$\frac{a+b+c}{a+b+c}$</td>
<td>(Rogers and Tanimoto, 1960)</td>
</tr>
<tr>
<td>20. Hamann</td>
<td>$\frac{a+b+c}{a+b+c}$</td>
<td>(Hamann, 1961)</td>
</tr>
<tr>
<td>21. Third Sokal-Sneath</td>
<td>$\frac{a+b+c}{a+b+c}$</td>
<td>(Sokal and Sneath, 1963)</td>
</tr>
<tr>
<td>22. Jaccard</td>
<td>$\frac{a+b+c}{a+b+c}$</td>
<td>(Jaccard, 1912)</td>
</tr>
<tr>
<td>23. First Kulczynsky</td>
<td>$\frac{a}{a+b}$</td>
<td>(Kulczynski, 1927)</td>
</tr>
<tr>
<td>24. Second Sokal-Sneath</td>
<td>$\frac{a+b+c}{a}$</td>
<td>(Sokal and Sneath, 1963)</td>
</tr>
<tr>
<td>25. Second Kulczynski</td>
<td>$\frac{a+b+c}{a}$</td>
<td>(Kulczynski, 1927)</td>
</tr>
<tr>
<td>26. Fourth Sokal-Sneath</td>
<td>$\frac{a+b+c}{a+b+c}$</td>
<td>(Kulczynski, 1927)</td>
</tr>
<tr>
<td>27. Odds ratio</td>
<td>$\frac{p(x</td>
<td>y)}{p(y</td>
</tr>
<tr>
<td>28. Yule's $\omega$</td>
<td>$\frac{ad-bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$</td>
<td>(Tan et al., 2002)</td>
</tr>
<tr>
<td>29. Yule's $Q$</td>
<td>$\frac{ad-bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$</td>
<td>(Tan et al., 2002)</td>
</tr>
<tr>
<td>30. Driver-Kroeber</td>
<td>$\sqrt{\frac{1}{a+b+c+d}}$</td>
<td>(Driver and Kroeber, 1932)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># name</th>
<th>formula</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>31. Fifth Sokal-Sneath</td>
<td>$\frac{a+d}{a+b+c+d}$</td>
<td>(Sokal and Sneath, 1963)</td>
</tr>
<tr>
<td>32. Pearson</td>
<td>$\frac{a-bc}{a+b+c}$</td>
<td>(Pearson, 1950)</td>
</tr>
<tr>
<td>33. Baroni-Urbani</td>
<td>$\frac{n_a}{n_a+n_b+n_c+n_d}$</td>
<td>(Baroni-Urbani and Buser, 1976)</td>
</tr>
<tr>
<td>34. Brauns-Blanquet</td>
<td>$\frac{a}{n}$</td>
<td>(Braun-Blanquet, 1932)</td>
</tr>
<tr>
<td>35. Simpson</td>
<td>$\frac{n_a+n_b+n_c+n_d}{n_a+n_b+n_c+n_d}$</td>
<td>(Simpson, 1943)</td>
</tr>
<tr>
<td>36. Michael</td>
<td>$\frac{a}{a+b+c}$</td>
<td>(Michael, 1920)</td>
</tr>
<tr>
<td>37. Mountford</td>
<td>$\frac{a}{a+b+c}$</td>
<td>(Kaufman and Rousseau, 1990)</td>
</tr>
<tr>
<td>38. Fager</td>
<td>$\frac{a}{b+c} - \frac{1}{2} \max(b, c)$</td>
<td>(Kaufman and Rousseau, 1990)</td>
</tr>
<tr>
<td>39. Unigram subtruples</td>
<td>$\log \frac{n_x}{n_y} - 3.29 \sqrt{\frac{n_x}{n_y} + 0.5 + \frac{1}{4} \cdot \frac{n_x}{n_y}}$</td>
<td>(Bahl et al. and Johnson, 2001)</td>
</tr>
<tr>
<td>40. U cost</td>
<td>$\log(1 + \frac{\text{max}(b, c) \cdot n_{xy}}{\text{max}(b, c) \cdot n_{xy} + \text{max}(b, c) \cdot n_{xy}})$</td>
<td>(Tulloss, 1997)</td>
</tr>
<tr>
<td>41. R cost</td>
<td>$\log(1 + \frac{\text{max}(b, c) \cdot n_{xy}}{\text{max}(b, c) \cdot n_{xy} + \text{max}(b, c) \cdot n_{xy}})$</td>
<td>(Tulloss, 1997)</td>
</tr>
<tr>
<td>42. T combined cost</td>
<td>$\sqrt{U \cdot S \cdot R}$</td>
<td>(Tulloss, 1997)</td>
</tr>
<tr>
<td>43. Kappa</td>
<td>$\frac{\text{predicted} \cdot \text{actual} - \text{expected}}{\text{predicted} \cdot \text{actual} + \text{expected}}$</td>
<td>(Tan et al., 2002)</td>
</tr>
<tr>
<td>44. J measure</td>
<td>$\text{max}[p(x</td>
<td>y) \cdot \log \frac{p(x</td>
</tr>
<tr>
<td>45. Gini index</td>
<td>$\text{max}[p(x</td>
<td>y) \cdot (p(y</td>
</tr>
<tr>
<td>46. Confidence</td>
<td>$\text{max}[p(y</td>
<td>x), p(x</td>
</tr>
<tr>
<td>47. Laplace</td>
<td>$\text{max}[\log(p(x</td>
<td>y)), \log(p(x</td>
</tr>
<tr>
<td>48. Conviction</td>
<td>$\text{max}[\log(p(x</td>
<td>y)), \log(p(x</td>
</tr>
<tr>
<td>49. Pietersky-Shapiro</td>
<td>$\frac{p(x</td>
<td>y)}{p(x)p(y)}$</td>
</tr>
<tr>
<td>50. Certainty factor</td>
<td>$\frac{p(x</td>
<td>y)}{p(x)p(y)}$</td>
</tr>
<tr>
<td>51. Collective strength</td>
<td>$\text{max}[p(x</td>
<td>y), p(x</td>
</tr>
<tr>
<td>52. Klossgen</td>
<td>$\sqrt{p(x</td>
<td>y) \cdot AV}$</td>
</tr>
</tbody>
</table>
Dimensionality Reduction

  - High dimensionality of word--document matrix
    - Sparsity
    - The order of rows and columns doesn’t matter
- Goal:
  - good similarity measure for words or documents
  - dense representation
- Sparse vs Dense vectors
  - Short vectors may be easier to use as features in machine learning (less weights to tune)
  - Dense vectors may generalize better than storing explicit counts
    - They may do better at capturing synonymy
    - In practice, they work better
Singular Value Decomposition (SVD)

- Solution idea:
  - Find a projection into a low-dimensional space (~300 dim)
  - That gives us a best separation between features

\[ \mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{V}^T \]

- \(\mathbf{U}\) is orthonormal
- \(\mathbf{D}\) is diagonal, sorted
Truncated SVD

We can approximate the full matrix by only considering the leftmost $k$ terms in the diagonal matrix (the $k$ largest singular values).

\[
A_{m \times n} \approx U_{m \times k} \Sigma_{k \times k} V_{k \times n}^T
\]

$k \ll m, n$
### Latent Semantic Analysis

<table>
<thead>
<tr>
<th>#0</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
</tr>
</thead>
<tbody>
<tr>
<td>we</td>
<td>music</td>
<td>company</td>
<td>how</td>
<td>program</td>
<td>10</td>
</tr>
<tr>
<td>said</td>
<td>film</td>
<td>mr</td>
<td>what</td>
<td>project</td>
<td>30</td>
</tr>
<tr>
<td>have</td>
<td>theater</td>
<td>its</td>
<td>about</td>
<td>russian</td>
<td>11</td>
</tr>
<tr>
<td>they</td>
<td>mr</td>
<td>inc</td>
<td>their</td>
<td>space</td>
<td>12</td>
</tr>
<tr>
<td>not</td>
<td>this</td>
<td>stock</td>
<td>or</td>
<td>russia</td>
<td>15</td>
</tr>
<tr>
<td>but</td>
<td>who</td>
<td>companies</td>
<td>this</td>
<td>center</td>
<td>13</td>
</tr>
<tr>
<td>be</td>
<td>movie</td>
<td>sales</td>
<td>are</td>
<td>programs</td>
<td>14</td>
</tr>
<tr>
<td>do</td>
<td>which</td>
<td>shares</td>
<td>history</td>
<td>clark</td>
<td>20</td>
</tr>
<tr>
<td>he</td>
<td>show</td>
<td>said</td>
<td>be</td>
<td>aircraft</td>
<td>sept</td>
</tr>
<tr>
<td>this</td>
<td>about</td>
<td>business</td>
<td>social</td>
<td>ballet</td>
<td>16</td>
</tr>
<tr>
<td>there</td>
<td>dance</td>
<td>share</td>
<td>these</td>
<td>its</td>
<td>25</td>
</tr>
<tr>
<td>you</td>
<td>its</td>
<td>chief</td>
<td>other</td>
<td>projects</td>
<td>17</td>
</tr>
<tr>
<td>are</td>
<td>disney</td>
<td>executive</td>
<td>research</td>
<td>orchestra</td>
<td>18</td>
</tr>
<tr>
<td>what</td>
<td>play</td>
<td>president</td>
<td>writes</td>
<td>development</td>
<td>19</td>
</tr>
<tr>
<td>if</td>
<td>production</td>
<td>group</td>
<td>language</td>
<td>work</td>
<td>21</td>
</tr>
</tbody>
</table>

[Deerwester et al., 1990]
Evaluation

- Intrinsic
- Extrinsic
- Qualitative

<table>
<thead>
<tr>
<th>WORD</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>...</th>
<th>d50</th>
</tr>
</thead>
<tbody>
<tr>
<td>summer</td>
<td>0.12</td>
<td>0.21</td>
<td>0.07</td>
<td>0.25</td>
<td>0.33</td>
<td>...</td>
<td>0.51</td>
</tr>
<tr>
<td>spring</td>
<td>0.19</td>
<td>0.57</td>
<td>0.99</td>
<td>0.30</td>
<td>0.02</td>
<td>...</td>
<td>0.73</td>
</tr>
<tr>
<td>fall</td>
<td>0.53</td>
<td>0.77</td>
<td>0.43</td>
<td>0.20</td>
<td>0.29</td>
<td>...</td>
<td>0.85</td>
</tr>
<tr>
<td>light</td>
<td>0.00</td>
<td>0.68</td>
<td>0.84</td>
<td>0.45</td>
<td>0.11</td>
<td>...</td>
<td>0.03</td>
</tr>
<tr>
<td>clear</td>
<td>0.27</td>
<td>0.50</td>
<td>0.21</td>
<td>0.56</td>
<td>0.25</td>
<td>...</td>
<td>0.32</td>
</tr>
<tr>
<td>blizzard</td>
<td>0.15</td>
<td>0.05</td>
<td>0.64</td>
<td>0.17</td>
<td>0.99</td>
<td>...</td>
<td>0.23</td>
</tr>
</tbody>
</table>
Extrinsic Evaluation

- Chunking
- POS tagging
- Parsing
- MT
- SRL
- Topic categorization
- Sentiment analysis
- Metaphor detection
- etc.
## Intrinsic Evaluation

- **WS-353** (Finkelstein et al. ‘02)
- **MEN-3k** (Bruni et al. ‘12)
- **SimLex-999 dataset** (Hill et al., 2015)

<table>
<thead>
<tr>
<th>word1</th>
<th>word2</th>
<th>similarity (humans)</th>
<th>similarity (embeddings)</th>
</tr>
</thead>
<tbody>
<tr>
<td>vanish</td>
<td>disappear</td>
<td>9.8</td>
<td>1.1</td>
</tr>
<tr>
<td>behave</td>
<td>obey</td>
<td>7.3</td>
<td>0.5</td>
</tr>
<tr>
<td>belief</td>
<td>impression</td>
<td>5.95</td>
<td>0.3</td>
</tr>
<tr>
<td>muscle</td>
<td>bone</td>
<td>3.65</td>
<td>1.7</td>
</tr>
<tr>
<td>modest</td>
<td>flexible</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>hole</td>
<td>agreement</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Spearman's rho (human ranks, model ranks)
Visualisation

- Visualizing Data using t-SNE (van der Maaten & Hinton’08)

Figure 6.5: Monolingual (top) and multilingual (bottom; marked with apostrophe) word projections of the antonyms (shown in red) and synonyms of “beautiful”.

[Faruqui et al., 2015]