Lexical and Vector Semantics

CSE354 - Spring 2021
Natural Language Processing
Tasks

- Word Sense Disambiguation
- Word Vectors
- Topic Modeling

how?

Traditionally:
- Probabilistic models
- Discriminant Learning: e.g. Logistic Regression
- Dimension Reduction: e.g. PCA)
Tasks

● Define common semantic tasks in NLP.
● Understand linguistic information necessary for semantic processing.
● Learn a couple approaches to semantic tasks.
● Motivate deep learning models necessary to capture language semantics.

Word Sense Disambiguation  
Word Vectors  
Topic Modeling  
Dependency Parsing  

how?

Traditionally:
○ Probabilistic models
○ Discriminant Learning: e.g. Logistic Regression
○ Transition-Based Parsing
○ Graph-Based Parsing

Current:
○ Recurrent Neural Network
○ Transformers
Preliminaries  (From SLP, Jurafsky et al., 2013)

Terminology: lemma and wordform

- **A lemma or citation form**
  - Same stem, part of speech, rough semantics
- **A wordform**
  - The inflected word as it appears in text

<table>
<thead>
<tr>
<th>Wordform</th>
<th>Lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>banks</td>
<td>bank</td>
</tr>
<tr>
<td>sung</td>
<td>sing</td>
</tr>
<tr>
<td>duermes</td>
<td>dormir</td>
</tr>
</tbody>
</table>
Lemmas have senses

- One lemma “bank” can have many meanings:

  Sense 1:  
  - “...a bank can hold the investments in a custodial account...”

  Sense 2:  
  - “…as agriculture burgeons on the east bank the river will shrink even more...”

- Sense (or word sense)
  - A discrete representation of an aspect of a word’s meaning.

- The lemma bank here has two senses
Preliminaries  (From SLP, Jurafsky et al., 2013)

- The lemma bank here has two senses.
Preliminaries  (From SLP, Jurafsky et al., 2013)

Homonymy

Homonyms: words that share a form but have unrelated, distinct meanings:

- bank$_1$: financial institution,  bank$_2$: sloping land
- bat$_1$: club for hitting a ball,  bat$_2$: nocturnal flying mammal

1. Homographs (bank/bank, bat/bat)
2. Homophones:
   1. Write and right
   2. Piece and peace
Preliminaries  (From SLP, Jurafsky et al., 2013)

Homonymy causes problems for NLP applications

- Information retrieval
  - “bat care”
- Machine Translation
  - bat: murciélago (animal) or bate (for baseball)
- Text-to-Speech
  - bass (stringed instrument) vs. bass (fish)
Word Sense Disambiguation

He put the port on the ship.

He walked along the port of the steamer.

He walked along the port next to the steamer.
He put the **port** on the ship.

He walked along the **port** of the steamer.

He walked along the **port** next to the steamer.
Word Sense Disambiguation

He put the **port** on the ship.

He walked along the **port** of the steamer.

He walked along the **port** next to the steamer.
He put the *port* on the ship.
He walked along the *port* of the steamer.
He walked along the *port* next to the steamer.

*port*. n.1 (a place (seaport or airport) where people and merchandise can enter or leave a country)

*port*. n.2 *port wine* (sweet dark-red dessert wine originally from Portugal)
He put the **port** on the ship.
He walked along the **port** of the steamer.
He walked along the **port** next to the steamer.

**port**.n.1 (a place (seaport or airport) where people and merchandise can enter or leave a country)

**port**.n.2 port wine (sweet dark-red dessert wine originally from Portugal)

**port**.n.3, embrasure, porthole (an opening (in a wall or ship or armored vehicle) for firing through)
He put the port on the ship.
He walked along the port of the steamer.
He walked along the port next to the steamer.

port.n.1 (a place (seaport or airport) where people and merchandise can enter or leave a country)

port.n.2 port wine (sweet dark-red dessert wine originally from Portugal)

port.n.3, embrasure, porthole (an opening (in a wall or ship or armored vehicle) for firing through)

larboard, port.n.4 (the left side of a ship or aircraft to someone who is aboard and facing the bow or nose)
He put the **port** on the ship.
He walked along the **port** of the steamer.
He walked along the **port** next to the steamer.

**port. n.1** (a place (seaport or airport) where people and merchandise can enter or leave a country)

**port. n.2** port wine (sweet dark-red dessert wine originally from Portugal)

**port. n.3**, embrasure, porthole (an opening (in a wall or ship or armored vehicle) for firing through)

larboard, **port. n.4** (the left side of a ship or aircraft to someone who is aboard and facing the bow or nose)

interface, **port. n.5** ((computer science) computer circuit consisting of the hardware and associated circuitry that links one device with another (especially a computer and a hard disk drive or other peripherals))
He put the **port** on the ship.
He walked along the **port** of the steamer.
He walked along the **port** next to the steamer.

As a verb...

1. **port** (put or turn on the left side, of a ship) "port the helm"
2. **port** (bring to port) "the captain ported the ship at night"
3. **port** (land at or reach a port) "The ship finally ported"
4. **port** (turn or go to the port or left side, of a ship) "The big ship was slowly porting"
5. **port** (carry, bear, convey, or bring) "The small canoe could be ported easily"
6. **port** (carry or hold with both hands diagonally across the body, especially of weapons) "port a rifle"
7. **port** (drink port) "We were porting all in the club after dinner"
8. **port** (modify (software) for use on a different machine or platform)

**port**.n.1 (a place (seaport or airport) where people and merchandise can enter or leave a country)
**port**.n.2 port wine (sweet dark-red dessert wine originally from Portugal)
**port**.n.3, embrasure, porthole (an opening (in a wall or ship or armored vehicle) for firing through)
**port**.n.4 (the left side of a ship or aircraft to someone who is aboard and facing the bow or nose)
**port**.n.5 ((computer science) computer circuit consisting of the hardware and associated circuitry that links one device with another (especially a computer and a hard disk drive or other peripherals))
Word Sense Disambiguation

A classification problem:

General Form:

\[ f\text{(sent\_tokens, (target\_index, lemma, POS))} \rightarrow \text{word\_sense} \]

He walked along the **port** next to the steamer.
Word Sense Disambiguation

A classification problem:

General Form:

\[ f(\text{sent\_tokens}, (\text{target\_index, lemma, POS})) \rightarrow \text{word\_sense} \]

Logistic Regression (or any discriminative classifier):

\[ P_{\text{lemma,POS}}(\text{sense} = s \mid \text{features}) \]

He walked along the port next to the steamer.
Word Sense Disambiguation

**General Form:**
\[ f(\text{sent_tokens}, (\text{target_index, lemma, POS})) \rightarrow \text{word_sense} \]

Logistic Regression (or any discriminative classifier):
\[ P(\text{lemma,POS} | \text{sense} = s | \text{features}) \]

He walked along the **port** next to the steamer. (Jurafsky, SLP 3)

---

Figure 19.3  The all-words WSD task, mapping from input words \((x)\) to WordNet senses \((y)\). Only nouns, verbs, adjectives, and adverbs are mapped, and note that some words (like **guitar** in the example) only have one sense in WordNet. Figure inspired by Chaplot and Salakhutdinov (2018).
Distributional Hypothesis:

Wittgenstein, 1945: “The meaning of a word is its use in the language”
Distributional Hypothesis:

Wittgenstein, 1945: “The meaning of a word is its use in the language”

Distributional hypothesis -- A word’s meaning is defined by all the different contexts it appears in (i.e. how it is “distributed” in natural language).

Firth, 1957: “You shall know a word by the company it keeps”

The nail hit the beam behind the wall.
The nail hit the beam behind the wall.
Approaches to WSD

I.e. how to operationalize the distributional hypothesis.

1. Bag of words for context
   E.g. multi-hot for any word in a defined “context”.

2. Surrounding window with positions
   E.g. one-hot per position relative to word).

3. Lesk algorithm
   E.g. compare context to sense definitions.

4. Selectors -- other target words that appear with same context
   E.g. counts for any selector.

5. Contextual Embeddings
   E.g. real valued vectors that “encode” the context (TBD).
Approaches to WSD

I.e. how to operationalize the distributional hypothesis.

1. Bag of words for context
   E.g. multi-hot for any word in a defined “context”.

2. Surrounding window with positions
   E.g. one-hot per position relative to word).

3. Lesk algorithm
   E.g. compare context to sense definitions.

4. Selectors -- other target words that appear with same context
   E.g. counts for any selector.

5. Contextual Embeddings
   E.g. real valued vectors that “encode” the context (TBD).
Lesk Algorithm for WSD

**Figure 19.10** The Simplified Lesk algorithm. The \texttt{COMPUTEOVERLAP} function returns the number of words in common between two sets, ignoring function words or other words on a stop list. The original Lesk algorithm defines the \textit{context} in a more complex way.

**Function** \texttt{SIMPLIFIED LESK}(word, sentence) \textbf{return}s best sense of word

\begin{verbatim}
best-sense \leftarrow \text{most frequent sense for word}
max-overlap \leftarrow 0
context \leftarrow \text{set of words in sentence}
for each sense in senses of word do
    signature \leftarrow \text{set of words in the gloss and examples of sense}
    overlap \leftarrow \text{COMPUTEOVERLAP}(signature, context)
    if overlap > max-overlap then
        max-overlap \leftarrow overlap
        best-sense \leftarrow sense
    end
return(best-sense)
\end{verbatim}
Lesk Algorithm for WSD

1. Bag of words for context
   E.g. multi-hot for any word in a defined "context".

2. Surrounding window with positions
   E.g. one-hot per position relative to word).

3. Lesk algorithm
   E.g. compare context to sense definitions.

4. Selectors -- other
   target words that appear with same context
   E.g. counts for any selector.

5. Contextual Embeddings
   E.g. real valued vectors that "encode" the context (TBD).

---

**bank.n.1** (sloping land (especially the slope beside a body of water)) "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"

**bank.n.2** (a financial institution that accepts deposits and channels the money into lending activities) "he cashed a check at the bank"; "that bank holds the mortgage on my home"

```python
overlap ← \text{COMPUTE}\text{OVERLAP}(\text{signature, context})

\text{if} \ overlap > \text{max-overlap} \ \text{then}
    \text{max-overlap} ← overlap
    \text{best-sense} ← \text{sense}
\text{end}

\text{return}(\text{best-sense})
```

---

*The bank can guarantee deposits will cover future tuition costs,* ...
Lesk Algorithm for WSD

I.e. how to operationalize the distributional hypothesis.

1. Bag of words for context
   E.g. multi-hot for any word in a defined "context".

2. Surrounding window with positions
   E.g. one-hot per position relative to word).

3. Lesk algorithm
   E.g. compare context to sense definitions.

4. Selectors -- other target words that appear with same context
   E.g. counts for any selector.

5. Contextual Embeddings
   E.g. real valued vectors that "encode" the context (TBD).

The **bank** can guarantee deposits will cover future tuition costs, ...
Lesk Algorithm for WSD

1. Bag of words for context
   E.g. multi-hot for any word in a defined "context".

2. Surrounding window with positions
   E.g. one-hot per position relative to word.

3. Lesk algorithm
   E.g. compare context to sense definitions.

4. Selectors -- other target words that appear with same context
   E.g. counts for any selector.

5. Contextual Embeddings
   E.g. real valued vectors that "encode" the context (TBD).

- **striker.n.1** (a forward on a soccer team)
- **striker.n.2** (someone receiving intensive training for a naval technical rating)
- **striker.n.3** (an employee on strike against an employer)
- **striker.n.4** (someone who hits) "a hard hitter"; "a fine striker of the ball"; "blacksmiths are good hitters"
- **striker.n.5** (the part of a mechanical device that strikes something)

\[
\text{overlap} \leftarrow \text{COMPUTEOVERLAP}(\text{signature}, \text{context})
\]

\[
\text{if overlap > max-overlap then}
\]

\[
\begin{align*}
\text{max-overlap} & \leftarrow \text{overlap} \\
\text{best-sense} & \leftarrow \text{sense}
\end{align*}
\]

\[
\text{end}
\]

\[
\text{return(best-sense)}
\]

He addressed the **strikers** at the rally.
Approaches to WSD

I.e. how to operationalize the distributional hypothesis.

1. Bag of words for context
   E.g. multi-hot for any word in a defined “context”.

2. Surrounding window with positions
   E.g. one-hot per position relative to word.

3. Lesk algorithm
   E.g. compare context to sense definitions.

4. Selectors -- other target words that appear with same context
   E.g. counts for any selector.

5. Contextual Embeddings
   E.g. real valued vectors that “encode” the context (TBD).
Approaches to WSD
I.e. how to operationalize the distributional hypothesis.

1. Bag of words for context
   E.g. multi-hot for any word in a defined “context”.

2. Surrounding window with positions
   E.g. one-hot per position relative to word.

3. Lesk algorithm
   E.g. compare context to sense definitions.

4. Selectors -- other target words that appear with same context
   E.g. counts for any selector.

5. Contextual Embeddings
   E.g. real valued vectors that “encode” the context (TBD).
Selectors

… a word which can take the place of another given word within the same local context (Lin, 1997)

Original version: Local context defined by dependency parse
Selectors

… a word which can take the place of another given word within the same local context (Lin, 1997)

Original version: Local context defined by dependency parse

He addressed the **strikers** at the rally.
Selectors

… a word which can take the place of another given word within the same local context (Lin, 1997)

Original version: Local context defined by dependency parse (Lin, 1997)

Web version: Local context defined by lexical patterns matched on the Web (Schwartz, 2008).

“He addressed the * at the rally.”
Selectors

... a word which can take the place of another given word within the same local context (Lin, 1997)

“..., but the bill now under discussion”
Selectors

Leverages hypernymy:
concept1 <is-a> concept2
"He addressed the strikers at the rally."
Why Are Selectors Effective?

Sets of selectors tend to vary extensively by word sense:

<table>
<thead>
<tr>
<th>bill-n.1</th>
<th>bill-n.2</th>
<th>bill-n.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>bill</td>
<td>bill</td>
<td>market</td>
</tr>
<tr>
<td>it</td>
<td>staff</td>
<td>system</td>
</tr>
<tr>
<td>legislation</td>
<td>system</td>
<td>paper</td>
</tr>
<tr>
<td>system</td>
<td>money</td>
<td>note</td>
</tr>
<tr>
<td>program</td>
<td>time</td>
<td>bill</td>
</tr>
<tr>
<td>law</td>
<td>it</td>
<td>bond</td>
</tr>
<tr>
<td>plan</td>
<td>tax</td>
<td>stock</td>
</tr>
<tr>
<td>you</td>
<td>work</td>
<td>debt</td>
</tr>
<tr>
<td>measure</td>
<td>rent</td>
<td>rate</td>
</tr>
<tr>
<td>project</td>
<td>tuition</td>
<td>report</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>occur-v.1</th>
<th>occur-v.2</th>
<th>occur-v.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>be</td>
<td>go</td>
<td>go</td>
</tr>
<tr>
<td>happen</td>
<td>get</td>
<td>look</td>
</tr>
<tr>
<td>occur</td>
<td>Come</td>
<td>break</td>
</tr>
<tr>
<td>go</td>
<td>have</td>
<td>remove</td>
</tr>
<tr>
<td>take</td>
<td>try</td>
<td>find</td>
</tr>
<tr>
<td>work</td>
<td>lead</td>
<td>get</td>
</tr>
<tr>
<td>come</td>
<td>listen</td>
<td>place</td>
</tr>
<tr>
<td>see</td>
<td>work</td>
<td>keep</td>
</tr>
<tr>
<td>have</td>
<td>be</td>
<td>stick</td>
</tr>
<tr>
<td>change</td>
<td>belong</td>
<td>stop</td>
</tr>
</tbody>
</table>
• Polls show wide, generalized support for some vague concept of service, but the **bill** now under discussion lacks any passionate public backing. training set never contained: “but the _ now under”

• … in his lecture, refers to the “startling experience which almost every person confesses, that particular passages of conversation and action have **occurred** to him in the same order before, whether dreaming or waking … small context is contradictory:
  “action have occurred” => occur-v.1 (“to happen or take place”)
  “occurred to him” => occur-v.2 (“to come to mind”)

<table>
<thead>
<tr>
<th>bill-n.1</th>
<th>bill-n.2</th>
<th>bill-n.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>bill</td>
<td>bill</td>
<td>market</td>
</tr>
<tr>
<td>it</td>
<td>staff</td>
<td>system</td>
</tr>
<tr>
<td>legislation</td>
<td>system</td>
<td>paper</td>
</tr>
<tr>
<td>system</td>
<td>money</td>
<td>note</td>
</tr>
<tr>
<td>program</td>
<td>time</td>
<td>bill</td>
</tr>
<tr>
<td>law</td>
<td>it</td>
<td>bond</td>
</tr>
<tr>
<td>plan</td>
<td>tax</td>
<td>stock</td>
</tr>
<tr>
<td>you</td>
<td>work</td>
<td>debt</td>
</tr>
<tr>
<td>measure</td>
<td>rent</td>
<td>rate</td>
</tr>
<tr>
<td>project</td>
<td>tuition</td>
<td>report</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>occur-v.1</th>
<th>occur-v.2</th>
<th>occur-v.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>be</td>
<td>go</td>
<td>go</td>
</tr>
<tr>
<td><strong>happen</strong></td>
<td>get</td>
<td>look</td>
</tr>
<tr>
<td>occur</td>
<td>Come</td>
<td>break</td>
</tr>
<tr>
<td>go</td>
<td>have</td>
<td>remove</td>
</tr>
<tr>
<td>take</td>
<td>try</td>
<td>find</td>
</tr>
<tr>
<td>work</td>
<td>lead</td>
<td>find</td>
</tr>
<tr>
<td>come</td>
<td>listen</td>
<td>place</td>
</tr>
<tr>
<td>see</td>
<td>work</td>
<td>keep</td>
</tr>
<tr>
<td>have</td>
<td>be</td>
<td>stick</td>
</tr>
<tr>
<td><strong>change</strong></td>
<td>belong</td>
<td>stop</td>
</tr>
</tbody>
</table>
## Supervised Selectors

<table>
<thead>
<tr>
<th></th>
<th>base</th>
<th>w/ sels</th>
<th>mfs</th>
<th>tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun</td>
<td>87.9</td>
<td>91.7</td>
<td>80.9</td>
<td>2559</td>
</tr>
<tr>
<td>verb</td>
<td>83.3</td>
<td>83.7</td>
<td>76.5</td>
<td>2292</td>
</tr>
<tr>
<td>both</td>
<td>85.7</td>
<td>87.9</td>
<td>78.8</td>
<td>4851</td>
</tr>
</tbody>
</table>

Accuracy over SemEval-2007: Task 17.
## Supervised Selectors

<table>
<thead>
<tr>
<th></th>
<th>base</th>
<th>w/ sels</th>
<th>mfs</th>
<th>tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun</td>
<td>87.9</td>
<td>91.7</td>
<td>80.9</td>
<td>2559</td>
</tr>
<tr>
<td>verb</td>
<td>83.3</td>
<td>83.7</td>
<td>76.5</td>
<td>2292</td>
</tr>
<tr>
<td>both</td>
<td>85.7</td>
<td>87.9</td>
<td>78.8</td>
<td>4851</td>
</tr>
</tbody>
</table>

Accuracy over SemEval-2007: Task 17.

<table>
<thead>
<tr>
<th></th>
<th>base</th>
<th>w/ sels</th>
<th>mfs</th>
<th>tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun</td>
<td>68.5</td>
<td>72.1</td>
<td>54.1</td>
<td>1766</td>
</tr>
<tr>
<td>verb</td>
<td>72.0</td>
<td>72.4</td>
<td>57.9</td>
<td>1927</td>
</tr>
<tr>
<td>adjective</td>
<td>49.4</td>
<td>53.4</td>
<td>54.7</td>
<td>148</td>
</tr>
<tr>
<td>all</td>
<td>69.4</td>
<td>71.5</td>
<td>56.1</td>
<td>3841</td>
</tr>
</tbody>
</table>

Accuracy over seneval-3 Lexical Sample. (fine-grained senses compared to SemEval)
More Background on WSD

https://prezi.com/m86pd1zbe_fy/?utm_campaign=share&utm_medium=copy

Covers a few approaches plus more background on “lexical semantics” in general.
Vector Semantics

1. Latent Semantic Analysis (LSA; Dimensionality Reduction-based Embeddings)
2. word2vec
3. Topic Modeling - Latent Dirichlet Allocation (LDA)
Vector Semantics

- Vectors which represent words or sequences
- Dimensionality Reduction
- Recurrent Neural Network and Sequence Models
Objective

To embed: convert a token (or sequence) to a vector that represents meaning.
Objective

To embed: convert a token (or sequence) to a vector that represents meaning, or is useful to perform downstream NLP application.
Objective

port \xrightarrow{\text{embed}} \mathbf{(\cdot)}
Objective

$\mathbf{port \xrightarrow{\text{embed}} \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}}$
Objective

one-hot is sparse vector

Port $\xrightarrow{\text{embed}}$

$$\begin{pmatrix}
0 \\
\ldots \\
0 \\
1 \\
\ldots \\
0
\end{pmatrix}$$

Prefer dense vectors
- Less parameters (weights) for machine learning model.
- May generalize better implicitly.
- May capture synonyms

For deep learning, in practice, they work better. Why? Roughly, less parameters becomes increasingly important when you are learning multiple layers of weights rather than just a single layer.
**Objective**

- Prefer dense vectors
  - Less parameters (weights) for machine learning model.
  - May generalize better implicitly.
  - May capture synonyms
  
For deep learning, in practice, they work better. Why? Roughly, less parameters becomes increasingly important when you are learning multiple layers of weights rather than just a single layer.

(one-hot is sparse vector)

(Jurafsky, 2012)
Objective

Prefer dense vectors

- Less parameters (weights) for machine learning model.
- May generalize better implicitly.
- May capture synonyms.

For deep learning, in practice, they work better. Roughly, less parameters becomes increasingly important when you are learning multiple layers of weights rather than just a single layer.

(one-hot is sparse vector)

(Jurafsky, 2012)
Objective

To embed: convert a token (or sequence) to a vector that represents meaning.

Wittgenstein, 1945: “The meaning of a word is its use in the language”

Distributional hypothesis -- A word’s meaning is defined by all the different contexts it appears in (i.e. how it is “distributed” in natural language).

Firth, 1957: “You shall know a word by the company it keeps”

The nail hit the beam behind the wall.
Word Vectors

"one-hot encoding"

beam $\xrightarrow{\text{embed}}$

\[
\begin{pmatrix}
0 \\
\ldots \\
0 \\
1 \\
\ldots \\
0
\end{pmatrix}
\]

Prefer dense vectors

- Less parameters (weights) for machine learning model.
- May generalize better implicitly.
- May capture synonyms

For deep learning, in practice, they work better. Why? Roughly, less parameters becomes increasingly important when you are learning multiple layers of weights rather than just a single layer.
The nail hit the beam behind the wall.
Objective

port embed

$$\begin{pmatrix}
0.53 \\
1.5 \\
3.21 \\
-2.3 \\
.76
\end{pmatrix}$$
**Objective**

**port**

- embed
- 0.53
- 1.5
- 3.21
- -2.3
- 0.76

**port.n.1** (a place (seaport or airport) where people and merchandise can enter or leave a country)

**port.n.2** port wine (sweet dark-red dessert wine originally from Portugal)

**port.n.3**, embrasure, porthole (an opening (in a wall or ship or armored vehicle) for firing through)

larboard, **port.n.4** (the left side of a ship or aircraft to someone who is aboard and facing the bow or nose)

interface, **port.n.5** (computer science) computer circuit consisting of the hardware and associated circuitry that links one device with another (especially a computer and a hard disk drive or other peripherals)
Dimensionality reduction
-- try to represent with only $p'$ dimensions

PCA-Based Embeddings
also known as "Latent Semantic Analysis"
PCA-Based Embeddings

also known as "Latent Semantic Analysis"
context words are features
w1, w2, w3, w4, ...

co-occurrence counts are cells.

target words are observations
w1, w2, w3, w4, ...

Dimensionality reduction
-- try to represent with only p’ dimensions
**PCA-Based Embeddings**

- Context words are features: $w_1, w_2, w_3, w_4, \ldots$  
- Target words are observations: $w_1, w_2, w_3, \ldots w_n$

Dimensionality reduction -- try to represent with only $p'$ dimensions: $p' < p$

Co-occurrence counts are cells.
Data (or, at least, what we want from the data) may be accurately represented with less dimensions.

Concept: Dimensionality Reduction in 3-D, 2-D, and 1-D

Data (or, at least, what we want from the data) may be accurately represented with less dimensions.

Concept: Dimensionality Reduction

Rank: Number of linearly independent columns of A.
(i.e. columns that can’t be derived from the other columns through addition).

Q: How many columns do we really need?

\[
\begin{pmatrix}
1 & -2 & 3 \\
2 & -3 & 5 \\
1 & 1 & 0 \\
\end{pmatrix}
\]
Concept: Dimensionality Reduction

Rank: Number of linearly independent columns of A. (i.e. columns that can’t be derived from the other columns through addition).

Q: How many columns do we really need?

\[
\begin{pmatrix}
1 & -2 & 3 \\
2 & -3 & 5 \\
1 & 1 & 0
\end{pmatrix}
\]

A: 2. The 1st is just the sum of the second two columns

\[
\begin{pmatrix}
1 \\
2 \\
1
\end{pmatrix} + \begin{pmatrix}
-2 \\
-3 \\
1
\end{pmatrix}
\]

... we can represent as linear combination of 2 vectors:
SVD-Based Embeddings

co-occurrence counts are cells.

c1, c2, c3, c4, …
cp'

context words are features
f1, f2, f3, f4, …
f_p

o1
o2
o3
...

Dimensionality reduction
-- try to represent with only p' dimensions

target words are observations
o_n
Dimensionality Reduction - PCA

Linear approximates of data in $r$ dimensions.

Found via *Singular Value Decomposition*:

$$X_{[n \times p]} = U_{[n \times r]} D_{[r \times r]} V_{[p \times r]}^T$$

$X$: original matrix, $U$: “left singular vectors”,
$D$: “singular values” (diagonal), $V$: “right singular vectors”
Dimensionality Reduction - PCA

Linear approximates of data in $r$ dimensions.

Found via **Singular Value Decomposition**:

$$X_{[n	imes p]} = U_{[n	imes r]} D_{[r	imes r]} V_{[p	imes r]}^T$$

- $X$: original matrix,
- $U$: “left singular vectors”,
- $D$: “singular values” (diagonal),
- $V$: “right singular vectors”
Dimensionality Reduction - PCA - Example

\[ X_{[n \times p]} = U_{[n \times r]} \ D_{[r \times r]} \ V_{[p \times r]}^T \]

Word co-occurrence counts:

\[
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2 \\
\end{pmatrix} \begin{pmatrix}
0.13 & 0.02 & -0.01 \\
0.41 & 0.07 & -0.03 \\
0.55 & 0.09 & -0.04 \\
0.68 & 0.11 & -0.05 \\
0.15 & -0.59 & 0.65 \\
0.07 & -0.73 & -0.67 \\
0.07 & -0.29 & 0.32 \\
\end{pmatrix} \begin{pmatrix}
12.4 & 0 & 0 \\
0 & 9.5 & 0 \\
0 & 0 & 1.3 \\
\end{pmatrix} \]

Dimensionality Reduction - PCA - Example

$$X_{[n \times p]} \cong U_{[n \times r]} D_{[r \times r]} V_{[p \times r]}^T$$

**Observation:**
- \(\text{count}(\text{beam}, \text{hit}) = 100\) -- horizontal dimension
- \(\text{count}(\text{beam}, \text{nail}) = 80\) -- vertical dimension
Dimensionality Reduction - PCA

Linear approximates of data in $r$ dimensions.

Found via *Singular Value Decomposition*:

$$X_{[n \times p]} \approx U_{[n \times r]} D_{[r \times r]} V_{[p \times r]}^T$$

$X$: original matrix, $U$: “left singular vectors”,
$D$: “singular values” (diagonal), $V$: “right singular vectors”

Projection (dimensionality reduced space) in 3 dimensions:

$$(U_{[n \times 3]} D_{[3 \times 3]} V_{[p \times 3]}^T)$$
Dimensionality Reduction - PCA

Linear approximates of data in \( r \) dimensions.

Found via *Singular Value Decomposition*:

\[
X_{[nxp]} \approx U_{[nxr]} D_{[rwx]} V_{[pxr]}^T
\]

**X**: original matrix,  \( \text{U}: \) “left singular vectors”,  
**D**: “singular values” (diagonal),  \( \text{V}: \) “right singular vectors”

To check how well the original matrix can be reproduced:

\[
Z_{[nxp]} = U D V^T, \text{ How does } Z \text{ compare to original } X?\]
Dimensionality Reduction - PCA

Linear approximates of data in \( r \) dimensions. Found via Singular Value Decomposition:

\[
X_{[nxp]} \cong U_{[nxr]}D_{[rxr]}V_{[p.xr]}^T
\]

- \( X \): original matrix,
- \( U \): “left singular vectors”,
- \( D \): “singular values” (diagonal),
- \( V \): “right singular vectors”

To check how well the original matrix can be reproduced:

\[
Z_{[nxp]} = U_{[nxr]}D_{[rxr]}V_{[p.xr]}^T,
\]

How does \( Z \) compare to original \( X \)?

The loss function that SVD solves

Goal: Minimize the sum of reconstruction errors:

\[
\sum_{i=1}^{N} \sum_{j=1}^{D} \|x_{ij} - z_{ij}\|^2
\]

- where \( x_{ij} \) are the “old” and \( z_{ij} \) are the “new” coordinates
Dimensionality Reduction - PCA

Linear approximates of data in \( r \) dimensions.

Found via \textit{Singular Value Decomposition}:

\[
X_{[n 	imes p]} \approx U_{[n 	imes r]} D_{[r 	imes r]} V_{[p 	imes r]}^T
\]

U, D, and V are unique

D: always positive
Objective

Prefer dense vectors

- Less parameters (weights) for machine learning model.
- May generalize better implicitly.
- May capture synonyms.

For deep learning, in practice, they work better. Roughly, less parameters becomes increasingly important when you are learning multiple layers of weights rather than just a single layer.

(one-hot is sparse vector) (Jurafsky, 2012)
Word2Vec

Principal: Predict missing word.

Similar to classification where $y = \text{context}$ and $x = \text{word}$.

$$p(\text{context} \mid \text{word})$$

To learn, maximize.
In practice, minimize

$$J = 1 - p(\text{context} \mid \text{word})$$
Word2Vec: Context

p(context | word)

2 Versions of Context:
1. Continuous bag of words (CBOW): Predict word from context
2. Skip-Grams (SG): predict context words from target
Word2Vec: Context

2 Versions of Context:
1. Continuous bag of words (CBOW): Predict word from context
2. Skip-Grams (SG): predict context words from target

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

(Jurafsky, 2017)
Word2Vec: Context

\[ p(\text{context} \mid \text{word}) \]

2. **Skip-Grams (SG):** predict context words from target

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

*The nail hit the beam behind the wall.*

(Jurafsky, 2017)
**Word2Vec: Context**

$p(\text{context} \mid \text{word})$

- **Continuous bag of words (CBOW):** Predict word from context
- **Skip-Grams (SG):** predict context words from target

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

$k$ negative examples ($y=0$) for every positive.

**How?** Randomly draw from unigram distribution

$$P(w) = \frac{\text{count}(w)}{\sum_{w} \text{count}(w)}$$

Example sentences:
- $x = (\text{hit}, \text{beam}), y = 1$
- $x = (\text{the}, \text{beam}), y = 1$
- $x = (\text{behind}, \text{beam}), y = 1$
- $x = (\text{happy}, \text{beam}), y = 0$
- $x = (\text{think}, \text{beam}), y = 0$

*The nail hit the beam behind the wall.*

(Jurafsky, 2017)
Word2Vec: Context

\[ p(\text{context} \mid \text{word}) \]

- Continuous bag of words (CBOW): Predict word from context
- Skip-Grams (SG): predict context words from target

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

\[ P_\alpha(w) = \frac{\text{count}(w)^\alpha}{\sum_w \text{count}(w)^\alpha} \]

\( \alpha = 0.75 \)

The nail hit the beam behind the wall.

\( x = (\text{hit, beam}), y = 1 \)
\( x = (\text{the, beam}), y = 1 \)
\( x = (\text{behind, beam}), y = 1 \)
... 
\( x = (\text{happy, beam}), y = 0 \)
\( x = (\text{think, beam}), y = 0 \)
...

\( k \) negative examples \((y=0)\) for every positive.

How? Randomly draw from unigram distribution adjusted:

\( \alpha = 0.75 \)

(Jurafsky, 2017)
Word2Vec: Context

\[ p(\text{context} \mid \text{word}) \]

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings

\( x = (\text{hit, beam}), y = 1 \)
\( x = (\text{the, beam}), y = 1 \)
\( x = (\text{behind, beam}), y = 1 \)
\( \ldots \)
\( x = (\text{happy, beam}), y = 0 \)
\( x = (\text{think, beam}), y = 0 \)
\( \ldots \)

\[ k \text{ negative examples (} y=0 \text{) for every positive.} \]

**How?** Randomly draw from unigram distribution adjusted:

\[ P_\alpha(w) = \frac{\text{count}(w)^\alpha}{\sum_w \text{count}(w)^\alpha} \]

\( \alpha = 0.75 \)

\( \alpha = 0.75 \)

\( \alpha = 0.75 \)

---

(Jurafsky, 2017)
Word2Vec: Context

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

The nail hit the beam behind the wall.

$P(y=1| c, t) = \frac{1}{1 + e^{-t \cdot c}}$

(Jurafsky, 2017)
Word2Vec: Context

- **Continuous bag of words (CBOW):** Predict word from context
- **Skip-Grams (SG):** predict context words from target

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

\[
\sigma(z) = \frac{1}{1 + e^{-z}}
\]

The nail hit the beam behind the wall.

\[
\begin{align*}
\text{single context:} & \quad P(y=1| \ c, \ t) = \frac{1}{1 + e^{-t \cdot c}} \\
\text{All Contexts} & \quad P(y=1| \ c, \ t) = \prod_{i=1}^{\kappa} \frac{1}{1 + e^{-t \cdot c_i}}
\end{align*}
\]

(Jurafsky, 2017)
Word2Vec: Context

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

\[ P(y=1 \mid c, t) = \frac{1}{1 + e^{-t \cdot c}} \]

**Intuition:** \( t \cdot c \) is a measure of similarity: 
\[ a \cdot b = ||a|| \cdot ||b|| \cdot \cos \theta \]
But, it is not a probability! To make it one, apply logistic activation:
\[ \sigma(z) = \frac{1}{1 + e^{-z}} \]

The nail hit the beam behind the wall.

(Jurafsky, 2017)
Word2Vec: Context

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

\[ P(y=1| c, t) = \frac{1}{1 + e^{-t \cdot c}} \]

\[ P(y=1| c, t) = \prod_{i=1}^{\kappa} \frac{1}{1 + e^{-t \cdot c_i}} \]

Intuition: \( t \cdot c \) is a measure of similarity:

\[ a \cdot b = \|a\| \|b\| \cos \theta \]

But, it is not a probability! To make it one, apply logistic activation:

\[ \sigma(z) = \frac{1}{1 + e^{-z}} \]
Word2Vec: How to Learn?

\[ P(y=1| c, t) \]

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

*The nail hit the beam behind the wall.*

(Jurafsky, 2017)
Word2Vec: How to Learn?

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

\[ P(y=1| c, t) \]

Assume 300 * |vocab| weights (parameters) for each of c and t.

The nail hit the beam behind the wall.

(Jurafsky, 2017)
Word2Vec: How to Learn?

\[ P(\text{y}=1 | c, t) \]
Assume 300 $\times$ |vocab| weights (parameters) for each of c and t
Start with random vectors (or all 0s)

1. Treat the target word and a neighboring context word as positive examples.
2. Randomly sample other words in the lexicon to get negative samples.
3. Use logistic regression to train a classifier to distinguish those two cases.
4. Use the weights as the embeddings.

*The nail hit the beam behind the wall.*

(Jurafsky, 2017)
Word2Vec: How to Learn?

\[ P(y=1|c, t) \]

Assume 300 * |vocab| weights (parameters) for each of c and t
Start with random vectors (or all 0s)

The nail hit the beam behind the wall.

(Jurafsky, 2017)
Word2Vec: How to Learn?

\[ P(y=1| c, t) \]

Assume 300 \* |vocab| weights (parameters) for each of c and t
Start with random vectors (or all 0s)

Goal:
Maximize similarity of (c, t) in positive data (y = 1)

The nail hit the beam behind the wall.

(Jurafsky, 2017)
Word2Vec: How to Learn?

\[ P(y=1| c, t) \]

Assume $300 \times |\text{vocab}|$ weights (parameters) for each of $c$ and $t$

Start with random vectors (or all 0s)

Goal:

- Maximize similarity of $(c, t)$ in positive data ($y = 1$)
- Minimize similarity of $(c, t)$ in negative data ($y = 0$)

The nail hit the beam behind the wall.
Word2Vec: How to Learn?

\[ P(y=1|c, t) \]

Assume \( 300 \times |\text{vocab}| \) weights (parameters) for each of \( c \) and \( t \)
Start with random vectors (or all 0s)

Goal:
Maximize similarity of \( (c, t) \) in positive data \( (y = 1) \)
Minimize similarity of \( (c, t) \) in negative data \( (y = 0) \)

\[
\sum_{(c,t)} (y) \log P(y = 1|c, t) + (y - 1) \log P(y = 0|c, t)
\]
Word2Vec: How to Learn?

\[ P(y=1|c, t) \]

Assume 300 * |vocab| weights (parameters) for each of c and t
Start with random vectors (or all 0s)

Goal:
Maximize similarity of (c, t) in positive data \((y = 1)\)
Minimize similarity of (c, t) in negative data \((y = 0)\)

\[
\sum_{(c,t)} (y) \log P(y = 1|c, t) + (y - 1) \log P(y = 0|c, t)
\]

\[
1 - P(y = 1|c, t) = \frac{e^{-t \cdot c}}{1 + e^{-t \cdot c}}
\]
Word2Vec: How to Learn?

Assume 300 * |vocab| weights (parameters) for each of c and t
Start with random vectors (or all 0s)

Goal:
Maximize similarity of (c, t) in positive data (y = 1)
Minimize similarity of (c, t) in negative data (y = 0)

\[ \sum_{(c,t)} (y) \log P(y = 1 | c, t) + (y - 1) \log P(y = 0 | c, t) \]

\[ 1 - P(y = 1 | c, t) = \frac{e^{-t \cdot c}}{1 + e^{-t \cdot c}} \]

Optimized using gradient descent type methods.
Word 2 Vec

\[
\sum_{(c,t)} (y) \log P(y = 1|c, t) + (y - 1) \log P(y = 0|c, t)
\]

(Jurafsky, 2017)
Word2Vec captures analogies (kind of)

(Jurafsky, 2017)
Word2Vec: Quantitative Evaluations

Compare to manually annotated pairs of words: WordSim-353 (Finkelstein et al., 2002)

Compare to words in context (Huang et al., 2012)

Answer TOEFL synonym questions.
Multi-class Loss Function

Logistic Regression Likelihood: \[ L(\beta_0, \beta_1, \ldots, \beta_k | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i} \]

Log Likelihood:
\[ \ell(\beta) = \sum_{i=1}^{N} y_i \log p(x_i) + (1-y_i) \log (1-p(x_i)) \]

Log Loss:
\[ J(\beta) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log p(x_i) + (1 - y_i) \log (1 - p(x_i)) \]
Multi-class Loss Function

Logistic Regression Likelihood: \[ L(\beta_0, \beta_1, ..., \beta_k | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i} \]

Log Likelihood:
\[
\ell(\beta) = \sum_{i=1}^{N} y_i \log p(x_i) + (1-y_i) \log (1-p(x_i))
\]

Log Loss:
\[
J(\beta) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log p(x_i) + (1-y_i) \log (1-p(x_i))
\]

Cross-Entropy Cost:
\[
J = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{|V|} y_{i,j} \log p(x_{i,j}) \quad (\text{a “multiclass” log loss})
\]
Multi-class Loss Function

Logistic Regression Likelihood: 
\[ L(\beta_0, \beta_1, \ldots, \beta_k | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i} \]

Log Likelihood:
\[ \ell(\beta) = \sum_{i=1}^{N} y_i \log p(x_i) + (1 - y_i) \log (1 - p(x_i)) \]

Log Loss:
\[ J(\beta) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log p(x_i) + (1 - y_i) \log (1 - p(x_i)) \]

Cross-Entropy Cost:
\[ J = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{|V|} y_{i,j} \log p(x_{i,j}) \] (a “multiclass” log loss)

In vector algebra form: 
\[ -\text{mean}(\text{sum}(y \times \log(y_{\text{pred}}))) \]
Tasks

- Word Sense Disambiguation
- Word Vectors
- Topic Modeling

how?

Traditionally:
- Probabilistic models
- Discriminant Learning: e.g. Logistic Regression
- Dimension Reduction: e.g. PCA)
Tasks

- Word Sense Disambiguation
- Word Vectors
- Topic Modeling

how?

Traditionally:
- Probabilistic models
- Discriminant Learning: e.g. Logistic Regression
- Dimension Reduction: e.g. PCA)
Topic Modeling

*Topic*: A group of highly related words and phrases. (aka "semantic field")

Example: from WTC responder interviews
(Son et al., 2021)
Topic Modeling

*Topic*: A group of highly related words and phrases. (aka "semantic field")
**Topic Modeling**

*Topic*: A group of highly related words and phrases. (aka "semantic field")
Select Example Topics
Generating Topics from Documents

- **Latent Dirichlet Allocation** -- a Bayesian probabilistic model where by words which appear in similar contexts (i.e. in essays that have similar sets of words) will be clustered into a prespecified number of topics.

- Rule of thumb: \[ |\text{topics}| = \frac{|\text{observations}|}{100} \]

- Each document receives a score per topic -- a probability: \( p(\text{topic}|\text{doc}) \).

<table>
<thead>
<tr>
<th>Doc 1</th>
<th>Doc 2</th>
<th>Doc 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>topic 1: .05</td>
<td>topic 1: .03</td>
<td>topic 1: .04</td>
</tr>
<tr>
<td>topic 2: .02</td>
<td>topic 2: .01</td>
<td>topic 2: .03</td>
</tr>
<tr>
<td>topic 3: .01</td>
<td>topic 3: .03</td>
<td>topic 3: .03</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>topic 100: .07</td>
<td>topic 100: .05</td>
<td>topic 100: .06</td>
</tr>
</tbody>
</table>
Latent Dirichlet Allocation
(Blei et al., 2003)

- LDA specifies a Bayesian probabilistic model where documents are viewed as a distribution of topics, and topics are a distribution of words.

**Observed:**
- $W$ -- observed word in document $m$

**Inferred:**
- $\theta$ -- topic distribution for document $m$
- $Z$ -- topic for word $n$ in document $m$
- $\varphi$ -- word distribution for topic $k$

**Priors**
- $\alpha$ -- parameter for Dirichlet prior on the topics per document.
- $\beta$ -- parameter for Dirichlet prior on the words per topic.
Latent Dirichlet Allocation
(Blei et al., 2003)

- LDA specifies a Bayesian probabilistic model where documents are viewed as a distribution of topics, and topics are a distribution of words.

- How to estimate (i.e. fit) the model parameters given data and priors? Common choices:
  ○ Gibb’s Sampling (best)
  ○ variational Bayesian Inference (fastest).

- Key Output: the "posterior" \( \phi = p(\text{word} \mid \text{topic}) \), the probability of a word given a topic.

  From this and \( p(\text{topic}) \), we can get: \( p(\text{topic} \mid \text{word}) \)

\[
p(\text{topic} \mid \text{doc}) = \sum_{\text{word} \in \text{topic}} p(\text{topic} \mid \text{word})p(\text{word} \mid \text{doc})
\]
Example

Most prevalent words for 4 topics are listed at the top and words associated with them from a Yelp review are colored accordingly below.

It depends what you look for in a hospital. Remember that this is a teaching hospital so you must adjust your expectations accordingly. This means many students who, bless their hearts, may ask you the same questions again and again. I waited for hours on standby to deliver my baby by emergency c-section. The kind nurses who served me during recovery and the anesthesiologist on duty during my surgery deserve praise. My OB was very competent, but I wish he were willing to do an extraversion or at least given me an epidural. I’m grateful they ultimately did what was best for my kid. However, I think things could have happened a lot more smoothly with better pain control. The only other thing to watch out for is your bills. This is the only institution I have been to that bills me prior to billing insurance. I fought two years to claim a credit through a database system change. The cafeteria gets flack for being all vegetarian but you just have to know what to order. Stay there for 1-2 weeks and you get the hang of what’s good and what’s not.

Topic Modeling Packages

Most Reliable: Mallet (Java; uses Gibb's Sampling), pymallet (slower than Mallet but high quality results)

Ease of use: Gensim (python; uses variational inference; implements word2vec as well)
Topic Modeling

Common applications:

- **Open vocabulary content analysis**: Describing the latent semantic categories of words or phrases present across a set of documents.

- **Embeddings for predictive task**: for all topics, use $p(\text{topic}|\text{document})$ as score. Feed to predictive model (e.g. classifier).
**Objective**

**port**

- **port.n.1** (a place (seaport or airport) where people and merchandise can enter or leave a country)
- **port.n.2** port wine (sweet dark-red dessert wine originally from Portugal)
- **port.n.3**, embrasure, porthole (an opening (in a wall or ship or armored vehicle) for firing through)
- **larboard**, **port.n.4** (the left side of a ship or aircraft to someone who is aboard and facing the bow or nose)
- **interface**, **port.n.5** ((computer science) computer circuit consisting of the hardware and associated circuitry that links one device with another (especially a computer and a hard disk drive or other peripherals))