Lecture 2: Distributional Semantics

Mirella Lapata

School of Informatics
University of Edinburgh
mlap@inf.ed.ac.uk

January 20, 2017
1 Vector Space Models
   - Distributional Hypothesis
   - Constructing Vector Spaces
   - Problems

2 Latent Semantic Analysis
   - Dimensionality Reduction
   - TOEFL Task

3 Discussion

Reading: J&M 19.1–19.5; Landauer and Dumais (1997).
The Meaning of “Bear”

- bear → carnivore
- predator, predatory animal
- animal, animate being, beast, brute, creature, fauna

WordNet: A lexical database for English
The Meaning of “Bear”

Latent Semantic Analysis (LSA; Landauer and Dumais, 1997)
Latent Dirichlet Allocation (LDA; Griffiths et al., 2007)
Neural Language Model (NLM; Collobert and Weston 2008)
The Meaning of “Bear”

Latent Semantic Analysis (LSA; Landauer and Dumais, 1997)
Latent Dirichlet Allocation (LDA; Griffiths et al., 2007)
Neural Language Model (NLM; Collobert and Weston 2008)
Vector Space Models
Latent Semantic Analysis
Discussion

The Meaning of “Bear”

Latent Semantic Analysis (LSA; Landauer and Dumais, 1997)
Latent Dirichlet Allocation (LDA; Griffiths et al., 2007)
Neural Language Model (NLM; Collobert and Weston 2008)
**Goal:** find a *representation* that succinctly describes the meaning of a word, phrase, sentence, document.
**Goal:** find a *representation* that succinctly describes the meaning of a word, phrase, sentence, document.

**Or at least:** find a representation so as to determine if two words or texts have *similar* meanings; linguistic environment $\approx$ corpus.
Motivation

**Goal:** find a *representation* that succinctly describes the meaning of a word, phrase, sentence, document.

**Or at least:** find a representation so as to determine if two words or texts have *similar* meanings; linguistic environment $\approx$ corpus.

**Why is this a good thing?**

- Retrieve documents relevant to a query.
- Use representations as features for entailment, sentiment, SRL, parsing, question answering, machine translation.
- Explain human learning and processing of words (word associations, speed of acquisition, etc).
- *Theory neutral*, few assumptions re word meaning.
Linguists have long conjectured that the context in which a word occurs determines its meaning:

- You shall know a word by the company it keeps (Firth);
- The meaning of a word is defined by the way it is used (Wittgenstein).

This leads to the distributional hypothesis about word meaning:

- the context surrounding a given word provides information about its meaning;
- words are similar if they share similar linguistic contexts;
- semantic similarity $\approx$ distributional similarity.
Distribution is represented using a *context vector*.

<table>
<thead>
<tr>
<th></th>
<th>pet</th>
<th>bone</th>
<th>fur</th>
<th>run</th>
<th>brown</th>
<th>screen</th>
<th>mouse</th>
<th>fetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$w_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$w_3$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Vector for $w$: all words that co-occur with $w$ (here, binary).
- Vector dimensions = number of context words.
- Similar words should have similar vectors.
Distribution is represented using a *context vector*.

<table>
<thead>
<tr>
<th></th>
<th>pet</th>
<th>bone</th>
<th>fur</th>
<th>run</th>
<th>brown</th>
<th>screen</th>
<th>mouse</th>
<th>fetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$w_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$w_3$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Vector for $w$: all words that co-occur with $w$ (here, binary).
- Vector dimensions = number of context words.
- Similar words should have similar vectors.
Constructing Vector Spaces

Words occur in context:

- car engine hood tires truck trunk
- car emissions hood make model trunk
- Chomsky corpus noun parsing tagging wonderful

Contexts can be obtained from corpora (large collections of text). Note that we have already removed stop words (frequent words such as *the*, *of*, *although*).
Constructing Vector Spaces

Select target words:

- Blue: car, engine, hood, tires, truck, trunk
- Red: car, emissions, hood, make, model, trunk
- Chomsky: corpus, noun, parsing, tagging, wonderful
Define the context (here: symmetric, $-5$, $+5$):

- **car** engine hood tires truck trunk
- **car** emissions **hood** make model trunk
- **Chomsky** corpus noun parsing tagging wonderful
Define the context (here: symmetric, −5, +5):

- car
- engine
- hood
- tires
- truck
- trunk

- car
- emissions
- hood
- make
- model
- trunk

- Chomsky
- corpus
- noun
- parsing
- tagging
- wonderful
Constructing Vector Spaces

Define the context (here: symmetric, $-5$, $+5$):

- **car**  engine  hood  tires  truck  trunk

- **car emissions**  hood  make  model  trunk

- **Chomsky**  corpus  noun  parsing  tagging  wonderful
Define the context (here: symmetric, $-5$, $+5$):

- **Car**
- **Engine**
- **Hood**
- **Tires**
- **Truck**
- **Trunk**

- **Car emissions**
- **Hood**
- **Make**
- **Model**
- **Trunk**

- **Chomsky**
- **Corpus**
- **Noun**
- **Parsing**
- **Tagging**
- **Wonderful**
### Constructing Vector Spaces

Create co-occurrence matrix:

<table>
<thead>
<tr>
<th></th>
<th>car</th>
<th>Chomsky</th>
<th>corpus</th>
<th>emissions</th>
<th>engine</th>
<th>hood</th>
<th>make</th>
<th>model</th>
<th>noun</th>
<th>parsing</th>
<th>tagging</th>
<th>tires</th>
<th>truck</th>
<th>trunk</th>
<th>wonderful</th>
</tr>
</thead>
<tbody>
<tr>
<td>car</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>hood</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Chomsky</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Informal algorithm for constructing vector spaces:

- pick the words you are interested in: *target words*;
- define number of words around target word: *context window*;
- count number of times target words co-occur with context words: *co-occurrence matrix*.

The context can also be defined in terms of documents, paragraphs, or sentences (rather than words around target word).
Constructing Vector Spaces

Measure the distance between vectors:

- **Euclidean**
- **Manhattan**
- **Cosine**
The *cosine* of the angle between two vectors $\mathbf{x}$ and $\mathbf{y}$ is:

\[
\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}
\]

The *Euclidean distance* of two vectors $\mathbf{x}$ and $\mathbf{y}$ is:

\[
\|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}
\]

Many more similarity measures exist.
Document similarity

- We represent document semantics also using vectors.
- **Bag-of-words (BOW) model**: order of words is irrelevant.
- Naive version: represent documents by word counts.

<table>
<thead>
<tr>
<th></th>
<th>pet</th>
<th>bone</th>
<th>fur</th>
<th>run</th>
<th>brown</th>
<th>screen</th>
<th>mouse</th>
<th>fetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$d_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$d_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

Document-term co-occurrence matrix
Using the Vector Space Model

Can compute similarities between documents, or between documents and queries.

Query: “computer pointer”
Pointwise Mutual Information

- Co-occurrence frequency is not a good measure of association.
- How often two words \( x \) and \( y \) occur compared with what we would expect if they were independent.

The **mutual information** between two random variables \( X \) and \( Y \)

\[
I(X, Y) = \sum_x \sum_y \log_2 \frac{P(x, y)}{P(x)P(y)}
\]

The **pointwise mutual information** between events \( x \) and \( y \)

\[
I(x, y) = \log_2 \frac{P(x, y)}{P(x)P(y)}
\]

Applied to co-occurrence vectors (\( w \) target word; \( c \) context word):

\[
PMI(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)}
\]
TF-IDF

\[ w_{t,d} = (1 + \log(tf_{t,d})) \log\left(\frac{N}{df_t}\right) \]

- \( tf_{t,d} \): frequency of word in document;
- \( idf \): \( N \) number of documents in collection; \( df_t \) number of documents in which term \( t \) occurs;
- Words that occur more frequently in a document are often more central to its meaning.
- Words that occur frequently in all documents have little semantic value.
- Increases with the number of occurrences within a document.
- Increases with the rarity of the term in the collection.
TF-IDF

\[ w_{t,d} = (1 + \log(tf_{t,d})) \log\left(\frac{N}{df_t}\right) \]

- \( tf_{t,d} \): frequency of word in document;
- \( df_t \): number of documents in which term \( t \) occurs;
- \( N \): number of documents in collection;

Words that occur more frequently in a document are often more central to its meaning.

Words that occur frequently in all documents have little semantic value.

Increases with the number of occurrences within a document.

Increases with the rarity of the term in the collection.
TF-IDF

\[ w_{t,d} = (1 + \log(tf_{t,d})) \log\left(\frac{N}{df_t}\right) \]

- \( tf_{t,d} \): frequency of word in document;
- idf: \( N \) number of documents in collection; \( df_t \) number of documents in which term \( t \) occurs;
- Words that occur more frequently in a document are often more central to its meaning.
- Words that occur frequently in all documents have little semantic value.
- Increases with the number of occurrences within a document.
- Increases with the rarity of the term in the collection.
The co-occurrence matrix can be very long and sparse (many zeros) and noisy (e.g., due to words with the same meaning).
The co-occurrence matrix can be very long and sparse (many zeros) and noisy (e.g., due to words with the same meaning).

- **auto** engine bonnet tires lorry boot
- **car** emissions hood make model trunk
- make hidden Markov model emissions normalize
The co-occurrence matrix can be very **long** and **sparse** (many zeros) and **noisy** (e.g., due to words with the same meaning).

- **auto**
- **engine**
- **bonnet**
- **tires**
- **lorry**
- **boot**

- **car**
- **emissions**
- **hood**
- **make**
- **model**
- **trunk**

- **make hidden**
- **Markov model**
- **emissions**
- **normalize**
The co-occurrence matrix can be very long and sparse (many zeros) and noisy (e.g., due to words with the same meaning).

- **auto**  engine  **bonnet**  tires  lorry  **boot**
- **car**  emissions  **hood**  make  model  **trunk**
- make  hidden  Markov  model  emissions  normalize
In order to address these problems, reduce the *dimensionality* of the co-occurrence matrix $A$:

- *project* the word vectors into a different subspace so that vector cosines more accurately represent semantic similarity;
- in a *lower dimensional space*, synonym vectors may not be orthogonal;
- *singular value decomposition* is a widely used projection method;
Dimensionality Reduction

- Projecting from two dimensions to one:
  - Single dimension (line) is chosen to follow direction of greatest variation.
  - Fit using least squares regression, i.e., minimizing

\[
\sum_{i=1}^{n} (y_i - f(x_i))^2
\]
The singular value decomposition of an $m$-by-$n$ matrix $A$ is:

$$A_{mn} = U_{mm} \Sigma_{mn} V_{nn}^T$$

- an orthogonal matrix $U$, a diagonal matrix $\Sigma$, and the transpose of an orthogonal matrix $V$.
- $m$-dimensional vectors making up the columns of $U$ are called left singular vectors.
- the $n$-dimensional vectors making up the columns of $V$ are called right singular vectors.
- $\Sigma$ contains the square roots of eigenvalues from $U$ or $V$ in descending order.
- A single value $A[i][j]$ in the matrix may be computed by the dot product of the $i$-th row vector and the $j$-th column vector, scaled by singular values.
Singular Value Decomposition (SVD)

\[ D = U \times \Sigma \times V^T \]

\[
\begin{bmatrix}
d_1 & d_2 & \cdots & d_n
\end{bmatrix} = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix}
\begin{bmatrix}
\sigma_1 & 0 & \cdots & 0 \\
0 & \sigma_2 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & \sigma_r
\end{bmatrix}
\begin{bmatrix}
v_1 \\
v_2 \\
\vdots \\
v_r
\end{bmatrix}
\]
Latent Semantic Analysis

- Best known vector space model (Landauer and Dumais, 1997).
- Natural language engineering:
  - lexicon acquisition (e.g., synonyms), unsupervised morphology;
  - essay grading, text coherence;
  - information retrieval;
  - language modeling, summarization, etc.
- Cognitive science: TOEFL 2nd language learning test.
The TOEFL Task

*Test of English as a Foreign Language* tests non-native speakers’ knowledge of English.

You will find the office at the main *intersection*.

(a) place  
(b) crossroads  
(c) roundabout  
(d) building

This is a standard task in the cognitive modeling literature, and vectors space models are frequently used to solve it.
The TOEFL Task

*Test of English as a Foreign Language* tests non-native speakers’ knowledge of English.

You will find the office at the main intersection.
(a) place
(b) crossroads
(c) roundabout
(d) building

This is a standard task in the cognitive modeling literature, and vectors space models are frequently used to solve it.
The TOEFL Task

- 80 items: 1 word/4 alternative words.
- Compute semantic representations for probe and answer words
- Word with largest cosine to the probe is correct answer.
- LSA was trained on a 4.6 M corpus from encyclopedia.
- LSA answered 64.4% items correctly.
- Non-native speakers’ average is 64.5%.
- This average is adequate for admission in many US universities.

What is the state of the art now?

Discussion

**Strengths:**
- fully automatic construction;
- representationally simple: all we need is a corpus and some notion of what counts as a word;
- language-independent, cognitively plausible.

**Weaknesses:**
- no underlying model, many ad-hoc parameters
- ambiguous words: their meaning is the average of all senses
- context words contribute indiscriminately to meaning;

The author received much acclaim for his new book. For author acclaim his much received new book.