Vector-based Models of Semantic Composition

Mirella Lapata and Jeff Mitchell

School of Informatics
University of Edinburgh

Seminar für Computerlinguistik, Heidelberg
Outline

1 Introduction
   - Semantic Space Models
   - Logic-based View
   - Connectionism

2 Composition Models

3 Evaluation
   - Phrase Similarity Task
   - Language Modeling

4 Conclusions
You shall know a word by the company it keeps (Firth, 1957).

- A word’s context provides information about its meaning.
- Words are similar if they share similar linguistic contexts.
- Distributional vs. semantic similarity.
Stuart B. Opotowsky was named vice president for this company with interests in insurance, tobacco, hotels and broadcasting.
Stuart B. Opotowsky was named vice president for this company with interests in insurance, tobacco, hotels and broadcasting.

- Select 2,000 most common content words as contexts.
Stuart B. Opotowsky was named vice president for this company with interests in insurance, tobacco, hotels and broadcasting.

- Select 2,000 most common content words as contexts.
- Five word context window each side of the target word.
Select 2,000 most common content words as contexts.

Five word context window each side of the target word.
A Simple Semantic Space

Select 2,000 most common content words as contexts.

Five word context window each side of the target word.
A Simple Semantic Space

- Select 2,000 most common content words as contexts.
- Five word context window each side of the target word.
- Convert counts to probabilities: $p(c|w)$.

<table>
<thead>
<tr>
<th></th>
<th>vice</th>
<th>president</th>
<th>tax</th>
<th>interests</th>
</tr>
</thead>
<tbody>
<tr>
<td>company</td>
<td>0.06</td>
<td>0.26</td>
<td>0.05</td>
<td>0.14</td>
</tr>
</tbody>
</table>
A Simple Semantic Space

- Select 2,000 most common content words as contexts.
- Five word context window each side of the target word.
- Convert counts to probabilities: $p(c|w)$.
- Divide through by probabilities of each context word: $\frac{p(c|w)}{p(c)}$. 

<table>
<thead>
<tr>
<th></th>
<th>vice</th>
<th>president</th>
<th>tax</th>
<th>interests</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>company</td>
<td>1.52</td>
<td>2.32</td>
<td>1.14</td>
<td>1.06</td>
<td>...</td>
</tr>
</tbody>
</table>
Select 2,000 most common content words as contexts.

Five word context window each side of the target word.

Convert counts to probabilities: \( p(c \mid w) \).

Divide through by probabilities of each context word: \( \frac{p(c \mid w)}{p(c)} \).

Cosine similarity: \( \text{sim}(\mathbf{w}_1, \mathbf{w}_2) = \frac{\mathbf{w}_1 \cdot \mathbf{w}_2}{\|\mathbf{w}_1\| \|\mathbf{w}_2\|} \).
An Alternative: Topic Models

**Key Idea:** documents are mixtures of topics, topics are probability distributions over words (Blei et al., 2003; Griffiths and Steyvers, 2002; 2003; 2004).

Topic models are *generative* and *structured*. For a new document:

1. Choose a distribution over topics
2. Choose a topic at random according to distribution
3. Draw a word from that topic

Statistical techniques used to invert the process: infer set of topics that were responsible for generating a collection of documents.
Probabilistic Generative Process

**TOPIC 1**
- bank
- loan
- money

**TOPIC 2**
- river
- bank
- stream

**DOC1**: money¹ bank¹ loan¹
   - bank¹ money¹ money¹
   - bank¹ loan¹

**DOC2**: money¹ bank¹
   - bank² river² loan¹ stream²
   - bank¹ money¹

**DOC3**: river² bank²
   - stream² bank² river² river²
   - stream² bank²
Statistical Inference

Figure 2.


### Meaning Representation

<table>
<thead>
<tr>
<th></th>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic n</th>
</tr>
</thead>
<tbody>
<tr>
<td>practical</td>
<td>0.39</td>
<td>0.02</td>
<td>...</td>
</tr>
<tr>
<td>difficulty</td>
<td>0.03</td>
<td>0.44</td>
<td>...</td>
</tr>
<tr>
<td>produce</td>
<td>0.06</td>
<td>0.17</td>
<td>...</td>
</tr>
</tbody>
</table>

- Topics are the dimensions of the space (500, 1000)
- Vector components: probability of word given topic
- Topics correspond to coarse-grained sense distinctions
- Cosine similarity can be used (probabilistic alternatives)

---

Mirella Lapata and Jeff Mitchell
Semantic space models are extremely popular across disciplines!

- Semantic Priming (Lund and Burgess, 1996)
- Text comprehension (Landauer and Dumais, 1997)
- Word association (McDonald, 2000)
- Information Retrieval (Salton et al., 1975)
- Thesaurus extraction (Grefenstette, 1994)
- Word Sense disambiguation (Schütze, 1998)
- Text Segmentation (Hirst, 1997)

**Automatic, language independent**
Semantic space models are extremely popular across disciplines!

- Semantic Priming (Lund and Burgess, 1996)
- Text comprehension (Landauer and Dumais, 1997)
- Word association (McDonald, 2000)
- Information Retrieval (Salton et al., 1975)
- Thesaurus extraction (Grefenstette, 1994)
- Word Sense disambiguation (Schütze, 1998)
- Text Segmentation (Hirst, 1997)
- **Automatic, language independent**

**Catch:** representation of the meaning of **single words**. What about **phrases** or **sentences**?
Quick Fix

It was not the sales manager who hit the bottle that day, but the office worker with the serious drinking problem.

That day the office manager, who was drinking, hit the problem sales worker with the bottle, but it was not serious.
Quick Fix

It was not the sales manager who hit the bottle that day, but the office worker with the serious drinking problem.

That day the office manager, who was drinking, hit the problem sales worker with the bottle, but it was not serious.

- Vector averaging: $p = \frac{1}{2}(u + v)$ (Foltz et al., 1998; Landauer et al., 1997); **syntax insensitive**
It was not the sales manager who hit the bottle that day, but the office worker with the serious drinking problem.

That day the office manager, who was drinking, hit the problem sales worker with the bottle, but it was not serious.

- Vector averaging: $\mathbf{p} = \frac{1}{2}(\mathbf{u} + \mathbf{v})$ (Foltz et al., 1998; Landauer et al., 1997); syntax insensitive
- Add a neighbor to the sum: $\mathbf{p} = \mathbf{u} + \mathbf{v} + \mathbf{n}$ (Kintsch, 2001); meaning of predicate depends on its argument
Meaning of whole is function of meaning of its parts (Frege, 1957).
Meaning of whole is function of meaning of its parts (Frege, 1957).

\[
\lambda u. \lambda v. \exists x (u @ x \land v @ x) \quad \lambda y. \textit{HORSE}(y) \quad \lambda z. \textit{RUN}(z)
\]
Logic-based View

Meaning of whole is function of meaning of its parts (Frege, 1957).

\[
\lambda u. \lambda v. \exists x (u@x \land v@x) \quad \lambda y. \text{HORSE}(y) \quad \lambda z. \text{RUN}(z)
\]

\[
\exists x (\text{HORSE}(x) \land \text{RUN}(x))
\]
Logic-based View

Meaning of whole is function of meaning of its parts (Frege, 1957).

\[ \lambda u.\lambda v.\exists x (u@x \land v@x) \quad \lambda y.\text{HORSE}(y) \quad \lambda z.\text{RUN}(z) \]

\[ \exists x (\text{HORSE}(x) \land \text{RUN}(x)) \]

- Logic can account for sentential meaning (Montague, 1974).
- Differences in meaning are qualitative rather than quantitative.
- Cannot express degrees of similarity.
Partee (1995): the meaning of the whole is a function of the meaning of the parts and of the way they are *syntactically* combined.
Compositionality

Partee (1995): the meaning of the whole is a function of the meaning of the parts and of the way they are **syntactically** combined.

Lakoff (1977): the meaning of the whole is a **greater** than the meaning of the parts.
Compositionality

Partee (1995): the meaning of the whole is a function of the meaning of the parts and of the way they are **syntactically** combined.

Lakoff (1977): the meaning of the whole is a **greater** than the meaning of the parts.

Frege (1884): never ask the meaning of a word in **isolation** but only **in the context** of a statement.
Compositionality

Partee (1995): the meaning of the whole is a function of the meaning of the parts and of the way they are **syntactically** combined.

Lakoff (1977): the meaning of the whole is a **greater** than the meaning of the parts.

Frege (1884): never ask the meaning of a word in **isolation** but only in **the context** of a statement.

Pinker (1994): composition of simple elements must allow the construction of **novel meanings** which go beyond those of the individual elements.
Tensor products: $p = u \otimes v$ (Smolensky, 1990); dimensionality
Tensor products: $\mathbf{p} = \mathbf{u} \otimes \mathbf{v}$ (Smolensky, 1990); **dimensionality**

Circular convolution: $\mathbf{p} = \mathbf{u} \otimes \mathbf{v}$ (Plate, 1991); **components are randomly distributed**
Tensor products: $\mathbf{p} = \mathbf{u} \otimes \mathbf{v}$ (Smolensky, 1990); **dimensionality**

Circular convolution: $\mathbf{p} = \mathbf{u} \odot \mathbf{v}$ (Plate, 1991); **components are randomly distributed**

Spatter codes: take the XOR of two vectors (Kanerva, 1998); **components are random bits**
A Framework for Semantic Composition

\[ p = f(u, v, R, K) \]
A Framework for Semantic Composition

\[ p = f(u, v, R, K) \]

composition of \( u, v \)
A Framework for Semantic Composition

\[ p = f(u, v, R, K) \]

composition of \( u, v \)

syntactic relationship

Assumptions:
1. Eliminate background knowledge \( K \)
2. Vary syntactic relationship \( R \)
3. \( p \) is in the same space as \( u \) and \( v \)
4. \( f() \) is a linear function of Cartesian product (additive model)
5. \( f() \) is a linear function of tensor product (multiplicative model)
A Framework for Semantic Composition

\[ p = f(u, v, R, K) \]

- composition of \( u, v \)
- syntactic relationship
- background knowledge

Assumptions:
1. Eliminate background knowledge \( K \)
2. Vary syntactic relationship \( R \)
3. \( p \) is in same space as \( u \) and \( v \)
4. \( f() \) is a linear function of Cartesian product (additive model)
5. \( f() \) is a linear function of tensor product (multiplicative model)
A Framework for Semantic Composition

\[ p = f(u, v, R) \]

- composition of \( u, v \)
- syntactic relationship

\textbf{Assumptions:}

1. Eliminate background knowledge \( K \)
A Framework for Semantic Composition

\[ p = f(u, v, OBJ) \]

Assumptions:
1. eliminate background knowledge \( K \)
2. vary syntactic relationship \( R \)
A Framework for Semantic Composition

\[ p = f(u, v, OBJ) \]

Assumptions:
1. eliminate background knowledge \( K \)
2. vary syntactic relationship \( R \)
3. \( p \) is in same space as \( u \) and \( v \)
A Framework for Semantic Composition

\[ p = f(u, v, OBJ) \]

composition of \( u, v \)

syntactic relationship

Assumptions:

1. eliminate background knowledge \( K \)
2. vary syntactic relationship \( R \)
3. \( p \) is in same space as \( u \) and \( v \)
4. \( f() \) is a linear function of Cartesian product (additive model)
A Framework for Semantic Composition

\[ p = f(u, v, OBJ) \]

composition of \( u, v \)

syntactic relationship

Assumptions:

1. eliminate background knowledge \( K \)
2. vary syntactic relationship \( R \)
3. \( p \) is in same space as \( u \) and \( v \)
4. \( f() \) is a linear function of Cartesian product (additive model)
5. \( f() \) is a linear function of tensor product (multiplicative model)
## Additive Models

\[ p = Au + Bv \]

## Instances

\[ p = u + v \]
\[ p = u + v + \sum_i n_i \]
\[ p = \alpha u + \beta v \]
\[ p = v \]
**Composition Models**

## Models

### Additive Models

\[ p = Au + Bv \]

### Instances

\[ p = u + v \]

\[ p = u + v + \sum n_i \]

\[ p = \alpha u + \beta v \]

\[ p = v \]

<table>
<thead>
<tr>
<th>practical</th>
<th>solution</th>
<th>economy</th>
<th>craft</th>
<th>create</th>
</tr>
</thead>
<tbody>
<tr>
<td>practical</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>difficulty</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>problem</td>
<td>2</td>
<td>15</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>

**practical + difficulty** = [1 14 6 14 4]
Models

Additive Models

\[ p = Au + Bv \]

Instances

\[ p = u + v \]

\[ p = u + v + \sum n_i \]

\[ p = \alpha u + \beta v \]

\[ p = v \]

<table>
<thead>
<tr>
<th></th>
<th>music</th>
<th>solution</th>
<th>economy</th>
<th>craft</th>
<th>create</th>
</tr>
</thead>
<tbody>
<tr>
<td>practical</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>difficulty</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>problem</td>
<td>2</td>
<td>15</td>
<td>7</td>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ \text{practical} + \text{difficulty} = [1 \ 14 \ 6 \ 14 \ 4] \]

\[ \text{practical} + \text{difficulty} + \text{problem} = [3 \ 29 \ 13 \ 23 \ 5] \]
Models

Additive Models

\[ p = Au + Bv \]

Instances

\[ p = u + v \]
\[ p = u + v + \sum_i n_i \]
\[ p = \alpha u + \beta v \]
\[ p = v \]

<table>
<thead>
<tr>
<th>music</th>
<th>solution</th>
<th>economy</th>
<th>craft</th>
<th>create</th>
</tr>
</thead>
<tbody>
<tr>
<td>practical</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>difficulty</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>problem</td>
<td>2</td>
<td>15</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>

\[
\text{practical} + \text{difficulty} = [1 14 6 14 4]
\]

\[
\text{practical} + \text{difficulty} + \text{problem} = [3 29 13 23 5]
\]

\[
0.4 \cdot \text{practical} + 0.6 \cdot \text{difficulty} = [0.6 5.6 3.2 6.4 1.6]
\]
Models

Additive Models

\[ p = Au + Bv \]

Instances

| practical | 0 | 6 | 2 | 10 | 4 |
| difficulty | 1 | 8 | 4 | 4 | 0 |
| problem | 2 | 15 | 7 | 9 | 1 |

\[ \text{practical} + \text{difficulty} = [1 14 6 14 4] \]

\[ \text{practical} + \text{difficulty} + \text{problem} = [3 29 13 23 5] \]

\[ 0.4 \cdot \text{practical} + 0.6 \cdot \text{difficulty} = [0.6 5.6 3.2 6.4 1.6] \]

\[ \text{difficulty} = [1 8 4 4 0] \]
### Multiplicative Models

\[ p = Cuv \]

#### Instances

\[ p = u \odot v \]
\[ p_i = u_i v_i \]

\[ p = u \otimes v \]
\[ p_{i,j} = u_i \cdot v_j \]

\[ p = u \oslash v \]
\[ p_i = \sum_j u_j \cdot v_{i-j} \]
Models

Multiplicative Models

\[ p = Cuv \]

Instances

\[ p = u \odot v \]
\[ p_i = u_i v_i \]

\[ p = u \otimes v \]
\[ p_{i,j} = u_i \cdot v_j \]

\[ p = u \oslash v \]
\[ p_i = \sum_j u_j \cdot v_{i-j} \]

<table>
<thead>
<tr>
<th>music</th>
<th>solution</th>
<th>economy</th>
<th>craft</th>
<th>create</th>
</tr>
</thead>
<tbody>
<tr>
<td>practical</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>difficulty</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

practical \odot difficulty = [0 48 8 40 0]
## Multiplicative Models

\[ p = C u v \]

### Instances

\[ p = u \odot v \]
\[ p_i = u_i v_i \]

\[ p = u \otimes v \]
\[ p_{i,j} = u_i \cdot v_j \]

\[ p = u \odot v \]
\[ p_i = \sum_j u_j \cdot v_{i-j} \]

<table>
<thead>
<tr>
<th>practical</th>
<th>music</th>
<th>solution</th>
<th>economy</th>
<th>craft</th>
<th>create</th>
</tr>
</thead>
<tbody>
<tr>
<td>difficulty</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\text{practical} \odot \text{difficulty} = [0 \ 48 \ 8 \ 40 \ 0]
\]

<table>
<thead>
<tr>
<th>practical</th>
<th>difficulty</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>music</td>
<td>solution</td>
<td>economy</td>
<td>craft</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>48</td>
<td>24</td>
<td>24</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>80</td>
<td>40</td>
<td>40</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>16</td>
<td>16</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Multiplicative Models

\[ p = Cuv \]

Instances

\[
\begin{align*}
p &= u \odot v \\
p_i &= u_i v_i \\
p &= u \otimes v \\
p_{i,j} &= u_i \cdot v_j \\
p &= u \oslash v \\
p_i &= \sum_j u_j \cdot v_{i-j}
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>music</th>
<th>solution</th>
<th>economy</th>
<th>craft</th>
<th>create</th>
</tr>
</thead>
<tbody>
<tr>
<td>practical</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>difficulty</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\text{practical} \odot \text{difficulty} = [0 \ 48 \ 8 \ 40 \ 0]
\]

\[
\text{practical} \otimes \text{difficulty} = \\
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
6 & 48 & 24 & 24 & 0 \\
2 & 16 & 8 & 8 & 0 \\
10 & 80 & 40 & 40 & 0 \\
4 & 32 & 16 & 16 & 0 \\
\end{bmatrix}
\]

\[
\text{practical} \oslash \text{difficulty} = [116 \ 50 \ 66 \ 62 \ 80]
\]
Dilation Models

\[ p = C \mathbf{uv} = \mathbf{Uv} \]
\[ U_{ij} = 0, \quad U_{ii} = u_i \]

\[ x = \frac{\mathbf{u} \cdot \mathbf{v}}{\mathbf{u} \cdot \mathbf{u}} \mathbf{u} \quad y = \mathbf{v} - x = \mathbf{v} - \frac{\mathbf{u} \cdot \mathbf{v}}{\mathbf{u} \cdot \mathbf{u}} \mathbf{u} \]

\[ \mathbf{v}' = \lambda x + y = (\lambda - 1) \frac{\mathbf{u} \cdot \mathbf{v}}{\mathbf{u} \cdot \mathbf{u}} \mathbf{u} + \mathbf{v} \]

\[ p = (\lambda - 1)(\mathbf{u} \cdot \mathbf{v})\mathbf{u} + (\mathbf{u} \cdot \mathbf{u})\mathbf{v} \]
Dilation Models

\[ p = C u v = U v \]
\[ U_{ij} = 0, \, U_{ii} = u_i \]

\[ x = \frac{u \cdot v}{u \cdot u} u \]
\[ y = v - x = v - \frac{u \cdot v}{u \cdot u} u \]

\[ v' = \lambda x + y = (\lambda - 1) \frac{u \cdot v}{u \cdot u} u + v \]

\[ p = (\lambda - 1)(u \cdot v)u + (u \cdot u)v \]
Dilation Models

\[ p = Cuv = Uv \]
\[ U_{ij} = 0, U_{ii} = u_i \]

\[ x = \frac{u \cdot v}{u \cdot u} u \quad \text{y} = v - x = v - \frac{u \cdot v}{u \cdot u} u \]

\[ v' = \lambda x + y = (\lambda - 1) \frac{u \cdot v}{u \cdot u} u + v \]

\[ p = (\lambda - 1)(u \cdot v)u + (u \cdot u)v \]
Dilation Models

\[ p = Cuv = Uv \]
\[ U_{ij} = 0, U_{ii} = u_i \]
\[ x = \frac{u \cdot v}{u \cdot u} u \quad y = v - x = v - \frac{u \cdot v}{u \cdot u} u \]
\[ v' = \lambda x + y = (\lambda - 1) \frac{u \cdot v}{u \cdot u} u + v \]
\[ p = (\lambda - 1)(u \cdot v)u + (u \cdot u)v \]
Dilation Models

\[ p = Cuv = Uv \]
\[ U_{ij} = 0, \quad U_{ii} = u_i \]

\[ x = \frac{u \cdot v}{u \cdot u} u \quad y = v - x = v - \frac{u \cdot v}{u \cdot u} u \]

\[ v' = \lambda x + y = (\lambda - 1) \frac{u \cdot v}{u \cdot u} u + v \]

\[ p = (\lambda - 1)(u \cdot v)u + (u \cdot u)v \]
Phrase Similarity Task

Originally proposed in Kintsch (2002):

- Elicit similarity judgments for adjective-noun, noun-noun, verb-object combinations.
- Phrase pairs from three bands: High, Medium, Low.
- Compute vectors for phrases, measure their similarity.
- Correlate model similarities with human ratings.
Phrase Similarity Task

Originally proposed in Kintsch (2002):

- Elicit similarity judgments for adjective-noun, noun-noun, verb-object combinations.
- Phrase pairs from three bands: High, Medium, Low.
- Compute vectors for phrases, measure their similarity.
- Correlate model similarities with human ratings.

<table>
<thead>
<tr>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>old person</td>
<td>kitchen door</td>
<td>produce effect</td>
</tr>
</tbody>
</table>
Phrase Similarity Task

Originally proposed in Kintsch (2002):

- Elicit similarity judgments for adjective-noun, noun-noun, verb-object combinations.
- Phrase pairs from three bands: High, Medium, Low.
- Compute vectors for phrases, measure their similarity.
- Correlate model similarities with human ratings.

<table>
<thead>
<tr>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>old person</td>
<td>elderly lady</td>
<td>small house</td>
</tr>
<tr>
<td>kitchen door</td>
<td>right hand</td>
<td></td>
</tr>
<tr>
<td>produce effect</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mirella Lapata and Jeff Mitchell
Phrase Similarity Task

Originally proposed in Kintsch (2002):

- Elicit similarity judgments for adjective-noun, noun-noun, verb-object combinations.
- Phrase pairs from three bands: High, Medium, Low.
- Compute vectors for phrases, measure their similarity.
- Correlate model similarities with human ratings.

<table>
<thead>
<tr>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>old person</td>
<td>elderly lady</td>
<td>small house</td>
</tr>
<tr>
<td>kitchen</td>
<td>bedroom window</td>
<td>housing department</td>
</tr>
<tr>
<td>door</td>
<td>produce effect</td>
<td></td>
</tr>
<tr>
<td>window</td>
<td>office worker</td>
<td></td>
</tr>
</tbody>
</table>
Phrase Similarity Task

Originally proposed in Kintsch (2002):

- Elicit similarity judgments for adjective-noun, noun-noun, verb-object combinations.
- Phrase pairs from three bands: High, Medium, Low.
- Compute vectors for phrases, measure their similarity.
- Correlate model similarities with human ratings.

<table>
<thead>
<tr>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>old person</td>
<td>elderly lady</td>
<td>small house</td>
</tr>
<tr>
<td>kitchen door</td>
<td>bedroom window</td>
<td>housing department</td>
</tr>
<tr>
<td>produce effect</td>
<td>achieve result</td>
<td>start work</td>
</tr>
<tr>
<td>effect</td>
<td>right hand</td>
<td></td>
</tr>
<tr>
<td></td>
<td>office worker</td>
<td></td>
</tr>
<tr>
<td></td>
<td>consider matter</td>
<td></td>
</tr>
</tbody>
</table>
Experimental Setup

Similarity Ratings
- 36 pairs (adj-noun, noun-noun, verb-noun) × 3 bands
  (324 pairs in total, created automatically, substitutability test)
- Ratings collected using Webexp (90 participants)
- Participants use 7-point similarity scale

Semantic Space
- Compare simple semantic space against LDA topic model
  (Blei et al. 2003)
- 2000 dimensions vs 100 topics, using cosine similarity measure
- Parameters for composition models tuned on dev set
## Results (for verb-obj)

<table>
<thead>
<tr>
<th>Model</th>
<th>Simple</th>
<th>LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>Kintsch</td>
<td>0.29</td>
<td>0.33</td>
</tr>
<tr>
<td>Weighted Additive</td>
<td>0.34</td>
<td>0.40</td>
</tr>
<tr>
<td>Multiplicative</td>
<td>0.37</td>
<td>0.34</td>
</tr>
<tr>
<td>Tensor Product</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>Circular Convolution</td>
<td>0.10</td>
<td>0.12</td>
</tr>
<tr>
<td>Dilation</td>
<td>0.38</td>
<td>0.41</td>
</tr>
<tr>
<td>Head Only</td>
<td>0.24</td>
<td>0.17</td>
</tr>
<tr>
<td>Humans</td>
<td>0.55</td>
<td></td>
</tr>
</tbody>
</table>

Multiplicative and dilation models best for simple space. Dilation and Additive models best for LDA model. Circular convolution is the worst performing model.
## Results (for verb-obj)

<table>
<thead>
<tr>
<th>Model</th>
<th>Simple</th>
<th>LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>Kintsch</td>
<td>0.29</td>
<td>0.33</td>
</tr>
<tr>
<td>Weighted Additive</td>
<td>0.34</td>
<td>0.40</td>
</tr>
<tr>
<td>Multiplicative</td>
<td>0.37</td>
<td>0.34</td>
</tr>
<tr>
<td>Tensor Product</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>Circular Convolution</td>
<td>0.10</td>
<td>0.12</td>
</tr>
<tr>
<td>Dilation</td>
<td>0.38</td>
<td>0.41</td>
</tr>
<tr>
<td>Head Only</td>
<td>0.24</td>
<td>0.17</td>
</tr>
<tr>
<td>Humans</td>
<td></td>
<td>0.55</td>
</tr>
</tbody>
</table>

Multiplicative and dilation models best for simple space. Dilation and additive models best for LDA model. Circular convolution is the worst performing model.
Results (for verb-obj)

<table>
<thead>
<tr>
<th>Model</th>
<th>Simple</th>
<th>LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>Kintsch</td>
<td>0.29</td>
<td>0.33</td>
</tr>
<tr>
<td>Weighted Additive</td>
<td>0.34</td>
<td>0.40</td>
</tr>
<tr>
<td><strong>Multiplicative</strong></td>
<td><strong>0.37</strong></td>
<td><strong>0.34</strong></td>
</tr>
<tr>
<td>Tensor Product</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>Circular Convolution</td>
<td>0.10</td>
<td>0.12</td>
</tr>
<tr>
<td><strong>Dilation</strong></td>
<td><strong>0.38</strong></td>
<td><strong>0.41</strong></td>
</tr>
<tr>
<td>Head Only</td>
<td>0.24</td>
<td>0.17</td>
</tr>
<tr>
<td>Humans</td>
<td>0.55</td>
<td></td>
</tr>
</tbody>
</table>

- Multiplicative and dilation models best for simple space
**Results (for verb-obj)**

<table>
<thead>
<tr>
<th>Model</th>
<th>Simple</th>
<th>LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>Kintsch</td>
<td>0.29</td>
<td>0.33</td>
</tr>
<tr>
<td>Weighted Additive</td>
<td>0.34</td>
<td>0.40</td>
</tr>
<tr>
<td>Multiplicative</td>
<td>0.37</td>
<td>0.34</td>
</tr>
<tr>
<td>Tensor Product</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>Circular Convolution</td>
<td>0.10</td>
<td>0.12</td>
</tr>
<tr>
<td>Dilation</td>
<td>0.38</td>
<td>0.41</td>
</tr>
<tr>
<td>Head Only</td>
<td>0.24</td>
<td>0.17</td>
</tr>
<tr>
<td>Humans</td>
<td>0.55</td>
<td></td>
</tr>
</tbody>
</table>

- Multiplicative and dilation models best for simple space
### Results (for verb-obj)

<table>
<thead>
<tr>
<th>Model</th>
<th>Simple</th>
<th>LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>Kintsch</td>
<td>0.29</td>
<td>0.33</td>
</tr>
<tr>
<td>Weighted Additive</td>
<td>0.34</td>
<td>0.40</td>
</tr>
<tr>
<td>Multiplicative</td>
<td>0.37</td>
<td>0.34</td>
</tr>
<tr>
<td>Tensor Product</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>Circular Convolution</td>
<td>0.10</td>
<td>0.12</td>
</tr>
<tr>
<td>Dilation</td>
<td>0.38</td>
<td>0.41</td>
</tr>
<tr>
<td>Head Only</td>
<td>0.24</td>
<td>0.17</td>
</tr>
<tr>
<td>Humans</td>
<td>0.55</td>
<td></td>
</tr>
</tbody>
</table>

- Multiplicative and dilation models best for simple space
- Dilation and Additive models best for LDA model
Results (for verb-obj)

<table>
<thead>
<tr>
<th>Model</th>
<th>Simple</th>
<th>LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>Kintsch</td>
<td>0.29</td>
<td>0.33</td>
</tr>
<tr>
<td>Weighted Additive</td>
<td>0.34</td>
<td>0.40</td>
</tr>
<tr>
<td>Multiplicative</td>
<td>0.37</td>
<td>0.34</td>
</tr>
<tr>
<td>Tensor Product</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>Circular Convolution</td>
<td>0.10</td>
<td>0.12</td>
</tr>
<tr>
<td>Dilation</td>
<td>0.38</td>
<td>0.41</td>
</tr>
<tr>
<td>Head Only</td>
<td>0.24</td>
<td>0.17</td>
</tr>
<tr>
<td>Humans</td>
<td>0.55</td>
<td></td>
</tr>
</tbody>
</table>

- Multiplicative and dilation models best for simple space
- Dilation and Additive models best for LDA model
## Results (for verb-obj)

<table>
<thead>
<tr>
<th>Model</th>
<th>Simple</th>
<th>LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additive</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>Kintsch</td>
<td>0.29</td>
<td>0.33</td>
</tr>
<tr>
<td>Weighted Additive</td>
<td>0.34</td>
<td>0.40</td>
</tr>
<tr>
<td>Multiplicative</td>
<td>0.37</td>
<td>0.34</td>
</tr>
<tr>
<td>Tensor Product</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>Circular Convolution</td>
<td>0.10</td>
<td>0.12</td>
</tr>
<tr>
<td>Dilation</td>
<td>0.38</td>
<td>0.41</td>
</tr>
<tr>
<td>Head Only</td>
<td>0.24</td>
<td>0.17</td>
</tr>
<tr>
<td>Humans</td>
<td>0.55</td>
<td></td>
</tr>
</tbody>
</table>

- Multiplicative and dilation models best for simple space
- Dilation and Additive models best for LDA model
- Circular convolution is worst performing model
Interim Summary

- General framework of semantic composition
- Different composition functions appropriate for different representations (additive vs. multiplicative)
- Dilation models overall best, syntax sensitive, parametric
- Results generalize to noun-noun, adj-noun, verb-obj combinations
Interim Summary

- General framework of semantic composition
- Different composition functions appropriate for different representations (additive vs. multiplicative)
- Dilation models overall best, syntax sensitive, parametric
- Results generalize to noun-noun, adj-noun, verb-obj combinations
- What are composition models good for?
Modeling Brain Activity

Tom Mitchell and collaborators
Wang et al., 2003; Mitchell et al., 2004; Mitchell et al., 2008; Hutchinson et al., 2009; Chang et al., 2009; Rustandi, 2009

- Can we observe differences in neural activity as people think about different concepts?
- Can we use vector-based models to explain observed neural activity?
Functional MRI

Monitors brain activity when people comprehend words or phrases. Measures changes related to blood flow and blood oxygenation.
Monitors brain activity when people comprehend words or phrases.
Functional MRI

Monitors brain activity when people comprehend words or phrases. Measures changes related to blood flow and blood oxygenation.
Functional MRI

soft bear

strong dog
Chang et al. (ACL, 2009)

- Participants see adjective-noun phrases
- Adjectives emphasize semantic properties of nouns
- Use vector-based models to account for variance in neural activity.
- Train regression model to fit activation profile of stimuli
- Multiplicative model outperforms non-compositional and additive model.
Interim Summary

- General framework of semantic composition
- Different composition functions appropriate for different representations (additive vs. multiplicative)
- Dilation models overall best, syntax sensitive, parametric
- Results generalize to noun-noun, adj-noun, verb-obj combinations
- What are composition models good for?
Interim Summary

- General framework of semantic composition
- Different composition functions appropriate for different representations (additive vs. multiplicative)
- Dilation models overall best, syntax sensitive, parametric
- Results generalize to noun-noun, adj-noun, verb-obj combinations

What are composition models good for?
- Modeling brain activity
Interim Summary

- General framework of semantic composition
- Different composition functions appropriate for different representations (additive vs. multiplicative)
- Dilation models overall best, syntax sensitive, parametric
- Results generalize to noun-noun, adj-noun, verb-obj combinations

What are composition models good for?
- modeling brain activity
- sentential priming, inductive inference
Interim Summary

- General framework of semantic composition
- Different composition functions appropriate for different representations (additive vs. multiplicative)
- Dilation models overall best, syntax sensitive, parametric
- Results generalize to noun-noun, adj-noun, verb-obj combinations
- **What are composition models good for?**
  - modeling brain activity
  - sentential priming, inductive inference
  - textual entailment, information retrieval, language modeling
Interim Summary

- General framework of semantic composition
- Different composition functions appropriate for different representations (additive vs. multiplicative)
- Dilation models overall best, syntax sensitive, parametric
- Results generalize to noun-noun, adj-noun, verb-obj combinations
- **What are composition models good for?**
  - modeling brain activity
  - sentential priming, inductive inference
  - textual entailment, information retrieval, **language modeling**
Language Modeling

What is the next word?
Language Modeling

What is the next word?

He is now president and chief operating officer.
Language Modeling

What is the next word?

He is now president and chief operating officer

‘chief operating’ is followed by ‘officer’ 99% of the time.
Language Modeling

What is the next word?

He is now president and chief operating officer

‘chief operating’ is followed by ‘officer’ 99% of the time.
Language Modeling

What is the next word?

He is now president and chief operating officer of the

‘of the’ is very frequent but not very predictive.
Language Modeling

What is the next word?

He is now president and chief operating officer of the

‘of the’ is very frequent but not very predictive.
Language Modeling

What is the next word?

He is now president and chief operating officer of the

Prior content indicative of domain the vocabulary is drawn from.
Language Modeling

What is the next word?

He is now president and chief operating officer of the

Prior content indicative of domain the vocabulary is drawn from.
Language Modeling

What is the next word?

He is now president and chief operating officer of the company.

Given semantic representations for ‘president’, ‘chief’, ‘operating’ and ‘officer’ how do we combine them to make the most predictive representation of this history?
Use vector composition in a language model as a way of capturing long-range dependencies.


How to combine vectors? How to construct them?

Focus on multiplicative and additive models.
He is now president and chief operating officer of the company.
He is now president and chief operating officer of the company

\[ p(\text{company}|\text{president}, \text{chief}, \text{operating}, \text{officer}) \]
A Language Model Based on Vector Composition

He is now president and chief operating officer of the company

\[ p(company|president, chief, operating, officer) \]

\[ p(w|h) = \text{sim}(w, h) \]
A Language Model Based on Vector Composition

He is now president and chief operating officer of the company

\[ p(company | president, chief, operating, officer) \]
\[ p(w|h) = \text{sim}(w, h) \]
\[ \text{sim}(w, h) \propto w \cdot h \]
A Language Model Based on Vector Composition

He is now president and chief operating officer of the company

\[ p(company | president, chief, operating, officer) \]
\[ p(w|h) = \text{sim}(w, h) \]
\[ \text{sim}(w, h) \propto w \cdot h = \sum w_i h_i \]
A Language Model Based on Vector Composition

He is now president and chief operating officer of the company

\[
p(\text{company} | \text{president, chief, operating, officer})
\]
\[
p(w | h) = \text{sim}(w, h)
\]
\[
\text{sim}(w, h) \propto w \cdot h = \sum \frac{p(c_i | w)}{p(c_i)} \frac{p(c_i | h)}{p(c_i)}
\]
A Language Model Based on Vector Composition

He is now president and chief operating officer of the company

\[
p(\text{company} | \text{president}, \text{chief}, \text{operating}, \text{officer})
\]

\[
p(w | h) = \text{sim}(w, h)
\]

\[
p(w | h) = p(w) \sum_i \frac{p(c_i | w)}{p(c_i)} \frac{p(c_i | h)}{p(c_i)} p(c_i)
\]
A Language Model Based on Vector Composition

He is now president and chief operating officer of the company

\[
p(company | president, chief, operating, officer) \\
\]

\[
p(w|h) = sim(w, h) \\
\]

\[
p(w|h) = p(w) \sum_i \frac{p(c_i|w)}{p(c_i)} \frac{p(c_i|h)}{p(c_i)} p(c_i) \\
\]

\[
h_n = f(w_n, h_{n-1}) \\
\]
He is now president and chief operating officer of the company

\[
p(\text{company} | \text{president}, \text{chief}, \text{operating}, \text{officer})
\]

\[
p(w | h) = \text{sim}(w, h)
\]

\[
p(w | h) = p(w) \sum_i \frac{p(c_i | w) p(c_i | h)}{p(c_i)} p(c_i)
\]

\[
h_n = f(w_n, h_{n-1})
\]

\[
h_1 = w_1
\]
Experimental Setup

- BLLIP Corpus
  - Training set - 38M words
  - Development set - 50K words
  - Test set - 50K words

- Numbers replaced with <NUM>
- Vocabulary of 20K word types
- Others replaced with <UNK>
- Perplexity of model predictions on test set
- Compare simple semantic space against LDA topic model
Integrating with an Ngram model

Linear interpolation

- $\lambda p_1(w) + (1 - \lambda)p_2(w)$
- But this will be most effective when models comparable in predictiveness.
Integrating with an Ngram model

**Linear interpolation**

- \( \lambda p_1(w) + (1 - \lambda) p_2(w) \)
- But this will be most effective when models comparable in predictiveness.

**Modify** \( p(w|h) \)
Integrating with an Ngram model

Linear interpolation

\[ \lambda p_1(w) + (1 - \lambda)p_2(w) \]

But this will be most effective when models comparable in predictiveness.

Modify \( p(w|h) \)

\[ p(w_n) \sum \frac{p(c_i|w_n)}{p(c_i)} \frac{p(c_i|h)}{p(c_i)} p(c_i) \]
Integrating with an Ngram model

Linear interpolation

- $\lambda p_1(w) + (1 - \lambda) p_2(w)$
- But this will be most effective when models comparable in predictiveness.

**Modify** $p(w|h)$

- $p(w_n) \sum \frac{p(c_i|w_n)}{p(c_i)} \frac{p(c_i|h)}{p(c_i)} p(c_i)$
- $p(w_n|w_{n-1}, w_{n-2}) \sum \frac{p(c_i|w_n)}{p(c_i)} \frac{p(c_i|h)}{p(c_i)} p(c_i)$
Integrating with an Ngram model

Linear interpolation

- \( \lambda p_1(w) + (1 - \lambda) p_2(w) \)
- But this will be most effective when models comparable in predictiveness.

Modify \( p(w|h) \)

- \( p(w_n) \sum \frac{p(c_i|w_n)}{p(c_i)} \frac{p(c_i|h)}{p(c_i)} p(c_i) \)
- \( p(w_n|w_{n-1}, w_{n-2}) \sum \frac{p(c_i|w_n)}{p(c_i)} \frac{p(c_i|h)}{p(c_i)} p(c_i) \)
Integrating with an Ngram model

Linear interpolation

\[ \lambda p_1(w) + (1 - \lambda)p_2(w) \]

But this will be most effective when models comparable in predictiveness.

Modify \( p(w|h) \)

\[ p(w_n) \sum \frac{p(c_i|w_n)}{p(c_i)} \frac{p(c_i|h)}{p(c_i)} p(c_i) \]

\[ p(w_n|w_{n-1}, w_{n-2}) \sum \frac{p(c_i|w_n)}{p(c_i)} \frac{p(c_i|h)}{p(c_i)} p(c_i) \]
Perplexities

Trigram

Perplexity

78.72
Perplexities

<table>
<thead>
<tr>
<th>Perplexity</th>
<th>Trigram</th>
<th>+Add SemSpace</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>78.72</td>
<td>76.66</td>
</tr>
</tbody>
</table>

Evaluation

Language Modeling
Perplexities

<table>
<thead>
<tr>
<th></th>
<th>78.72</th>
<th>76.66</th>
<th>75.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigram</td>
<td>+Add SemSpace</td>
<td>+Multiply SemSpace</td>
<td></td>
</tr>
</tbody>
</table>
Perplexities

<table>
<thead>
<tr>
<th>Method</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigram</td>
<td>78.72</td>
</tr>
<tr>
<td>+Add SemSpace</td>
<td>76.66</td>
</tr>
<tr>
<td>+Multiply SemSpace</td>
<td>75.01</td>
</tr>
<tr>
<td>+Add LDA</td>
<td>76.60</td>
</tr>
</tbody>
</table>
Perplexities

<table>
<thead>
<tr>
<th>Model</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigram</td>
<td>78.72</td>
</tr>
<tr>
<td>+Add SemSpace</td>
<td>76.66</td>
</tr>
<tr>
<td>+Multiply SemSpace</td>
<td>75.01</td>
</tr>
<tr>
<td>+Add LDA</td>
<td>78.60</td>
</tr>
<tr>
<td>+Multiply LDA</td>
<td>123.95</td>
</tr>
</tbody>
</table>
Perplexities

![Bar chart showing perplexity values for different models.
- Trigram: 78.72
- Trigram + Add SemSpace: 76.66
- Trigram + Multiply SemSpace: 75.01
- Trigram + Add LDA: 78.60
- Trigram + Multiply LDA: 123.95]
Comparison to Parsing

- Model incorporates semantic dependencies into a trigram model.
- Increases the probability of upcoming words which are semantically similar to the history.
- Syntactic information also captures long-range dependencies.
- Language models based on syntactic structure.
- Interpolate composition models with Roark’s (2001) parser.
Perplexities

<table>
<thead>
<tr>
<th>Trigram</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>78.72</td>
</tr>
</tbody>
</table>
Perplexities

<table>
<thead>
<tr>
<th>Trigram + Parser</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>78.72</td>
</tr>
<tr>
<td></td>
<td>75.22</td>
</tr>
</tbody>
</table>
Perplexities

Mirella Lapata and Jeff Mitchell
Perplexities

- Trigram: 78.72
- +Parser: 75.22
- +Add SemSpace: 73.45
Perplexities

<table>
<thead>
<tr>
<th>Method</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigram + Parser</td>
<td>78.72</td>
</tr>
<tr>
<td>+Add SemSpace</td>
<td>75.22</td>
</tr>
<tr>
<td>+Multiply SemSpace</td>
<td>73.45</td>
</tr>
<tr>
<td></td>
<td>71.32</td>
</tr>
</tbody>
</table>
Perplexities

The graph shows the perplexity values for different language models:

- Trigram
- Trigram + Parser
- Trigram + Add SemSpace
- Trigram + Multiply SemSpace
- Trigram + Add LDA

The perplexity values are as follows:

- Trigram: 78.72
- Trigram + Parser: 75.22
- Trigram + Add SemSpace: 73.45
- Trigram + Multiply SemSpace: 71.32
- Trigram + Add LDA: 71.58
Perplexities

![Bar chart showing perplexities for different models. The models include Trigram, Trigram + Parser, Trigram + Add, Trigram + Multiply, Trigram + Add SemSpace, Trigram + Multiply SemSpace, Trigram + Add LDA, Trigram + Multiply LDA. The perplexity values are 78.72, 75.22, 73.45, 71.32, 71.58, 87.93.]
Perplexities

<table>
<thead>
<tr>
<th>Model</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigram + Parser</td>
<td>78.72</td>
</tr>
<tr>
<td>+ Add SemSpace</td>
<td>75.22</td>
</tr>
<tr>
<td>+ Multiply SemSpace</td>
<td>73.45</td>
</tr>
<tr>
<td>+ Add LDA</td>
<td>71.32</td>
</tr>
<tr>
<td>+ Multiply LDA</td>
<td>71.58</td>
</tr>
<tr>
<td></td>
<td>87.93</td>
</tr>
</tbody>
</table>
Work so far

- Vector composition for phrase similarity and language modeling
- Compared a simple semantic space to LDA
- Different composition functions appropriate for each model
- Semantic dependencies complementary to syntactic ones

Future work

- Incorporate syntax into composition (parser that outputs a compositional vector-based representation of a sentence)
- Optimize vectors and composition function on specific tasks
LDA Topics

![Graph showing perplexity vs number of topics for additive and multiplicative methods. The graph illustrates how perplexity decreases as the number of topics increases for both methods, with the additive method showing a sharper drop at lower topic numbers and a relatively flat line for higher numbers.]