Language Representation and Modeling

Kapil Thadani
kapil@cs.columbia.edu

YAHOO!
RESEARCH
Schedule

○ Language representation and modeling ......................... March 1
  - Sparse and distributional representations
  - word2vec and doc2vec
  - Task-specific representations
  - Language modeling

○ Encoder-decoder frameworks .................................................. March 8
  - RNN units: LSTMs, GRUs
  - Sequence-to-sequence models
  - Attention mechanism
  - Copying mechanism
  - Scheduled sampling

○ Applications ................................................................. March 22
  - Parsing
  - Question-answering
  - Entailment
  - Machine reading
  - Dialog systems
Outline

○ Language representation
  - Bag of words
  - Distributional hypothesis
  - word2vec
  - doc2vec

○ Task-specific representations
  - Convolutional NNs
  - Recurrent NNs
  - Recursive NNs

○ Language modeling
  - Probabilistic and discriminative LMs
  - NNLM
  - RNNLMs
Formal language

(i) Set of sequences over symbols from an alphabet

- sentences
- words
- vocabulary
- utterances
- morphemes
- documents
- MWEs

(ii) Rules for valid sequences

- spelling
- orthography, morphology
- grammar
- syntax
- meaning
- semantics, discourse, pragmatics, ···
Natural language

(i) Set of sequences over symbols from an alphabet

- sentences
- words
- vocabulary
- utterances
- morphemes
- documents
- MWEs

(ii) Rules for valid sequences

- spelling
- orthography, morphology
- grammar
- syntax
- meaning
- semantics, discourse, pragmatics, ···
Lexical semantics

dog
Lexical semantics

- **Hypernymy**
  - dog
  - mammal
  - pet
  - canine

- **Meronymy**
  - dog
  - poodle
  - puppy
  - paw

- **Synonymy**
  - dog
  - canine

- **Holonymy**
  - dog
  - pack
Lexical semantics

- **mammal**
- **pet**
- **canine**
- **cat**
- **bark**
- **meronymy**
- **holonymy**
- **hyponymy**
- **synonymy**
- **opposition**
- **co-occurrence**
- **slang**
- **leash**
- **co-occurrence**
- **paw**
- **dawg**
- **poodle**
- **puppy**
Lexical semantics

- **Dog**
  - Hypernymy: Mammal, Pet
  - Hyponymy: Poodle, Puppy
  - Meronymy: Paw
  - Holonymy: Pack
  - Co-occurrence: Leash, Bark
  - Slang: Dawg
  - Opposition: Cat
  - Synonymy: Canine
  - Polysemy: Wretch, Frankfurter
  - Verb: Aggravate, Shadow
  - Name: "Dog the Bounty Hunter"
Sparse representations

Word: one-hot (1-of-$V$) vectors
Document: "bag of words"

Emphasize rare words with inverse document frequency (IDF)

Compare documents with cosine similarity

+ Simple and interpretable
  - No notion of word order
  - No implicit semantics
  - Curse of dimensionality with large $|V|$
Distributional approaches

Words that occur in similar contexts have similar meanings

e.g., record word co-occurrence within a context window over a large corpus

Weight association with pointwise mutual information (PMI), etc

\[
PMI(w_1, w_2) = \log_2 \frac{p(w_1, w_2)}{p(w_1)p(w_2)}
\]
Latent Semantic Analysis

Construct term-document matrix

\[ M = \begin{pmatrix} w_1^{(1)} & w_1^{(2)} & \cdots \\ w_2^{(1)} & \cdots & \\ \vdots & \vdots & \ddots \end{pmatrix} \]

Singular value decomposition

\[ M \approx u_1 u_2 u_3 \cdots \lambda_1 \lambda_2 \lambda_3 \cdots \]

Select top \( k \) singular vectors for \( k \)-dim embeddings of words/docs
word2vec

Continuous Bag-of-Words (CBOW)
  - Predict target $w_t$ given context $w_{t-c}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c}$

$$
\ell(W, U) = -\log p(w_t|w_{t-c} \cdots w_{t+c})
$$

$$
p(w_j|w_{t-c} \cdots w_{t+c}) = \frac{e^{U_jh_t}}{\sum_k e^{U_kh_t}}
$$

$$
h_t = \frac{1}{2c} \sum_{i=0}^{c} W^\top w_{t+i}
$$

Input

Projection (averaged)

Softmax

Loss

Label
word2vec

Skip-gram

- Predict context $w_{t-c}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c}$ given target $w_t$

$$
p(w_j|w_t) = \frac{e^{U_j h_t}}{\sum_k e^{U_k h_t}}
$$

$$
h_t = W^T w_t
$$

$$
\ell(W, U) = -\sum_{i=-c}^{c} \log p(w_{t+i}|w_t)
$$
Cost of computing $\nabla p(w_j | \cdots)$ is proportional to $V$!

**Alternative 1: Hierarchical softmax**
- Predict path in binary tree representation of output layer
- Reduces to $\log_2(V)$ binary decisions

$$p(w_t = \text{“dog”} | \cdots) = (1 - \sigma(U_{0}h_t)) \times \sigma(U_{1}h_t) \times \sigma(U_{4}h_t)$$
Cost of computing $\nabla p(w_j | \cdots)$ is proportional to $V$!

Alternative 2: Negative sampling

- Change objective to differentiate target vector from noisy samples with logistic regression

$$\max \log \sigma(u_j^\top h_t) + \sum_{k=1}^{K} \mathbb{E}_{w_m \sim \Psi} \log \sigma(-u_m^\top h_t)$$

where $u_j = U_j = j$’th column of $U$

and $w_j \in \text{context}(w_t)$

- Noise distribution $\Psi$ typically unigram, uniform or in between
- Number of samples $K$ typically 5–20
word2vec

Linear relationships between related words

Mikolov et al. (2013)
Country and Capital Vectors Projected by PCA

Visualizing lexical relationships

word2vec

Mikolov et al. (2013)
### word2vec

Mikolov et al. (2013)

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>France - Paris</td>
<td>Italy: Rome</td>
<td>Japan: Tokyo</td>
<td>Florida: Tallahassee</td>
</tr>
<tr>
<td>big - bigger</td>
<td>small: larger</td>
<td>cold: colder</td>
<td>quick: quicker</td>
</tr>
<tr>
<td>Miami - Florida</td>
<td>Baltimore: Maryland</td>
<td>Dallas: Texas</td>
<td>Kona: Hawaii</td>
</tr>
<tr>
<td>Einstein - scientist</td>
<td>Messi: midfielder</td>
<td>Mozart: violinist</td>
<td>Picasso: painter</td>
</tr>
<tr>
<td>Sarkozy - France</td>
<td>Berlusconi: Italy</td>
<td>Merkel: Germany</td>
<td>Koizumi: Japan</td>
</tr>
<tr>
<td>copper - Cu</td>
<td>zinc: Zn</td>
<td>gold: Au</td>
<td>uranium: plutonium</td>
</tr>
<tr>
<td>Berlusconi - Silvio</td>
<td>Sarkozy: Nicolas</td>
<td>Putin: Medvedev</td>
<td>Obama: Barack</td>
</tr>
<tr>
<td>Microsoft - Windows</td>
<td>Google: Android</td>
<td>IBM: Linux</td>
<td>Apple: iPhone</td>
</tr>
<tr>
<td>Microsoft - Ballmer</td>
<td>Google: Yahoo</td>
<td>IBM: McNealy</td>
<td>Apple: Jobs</td>
</tr>
<tr>
<td>Japan - sushi</td>
<td>Germany: bratwurst</td>
<td>France: tapas</td>
<td>USA: pizza</td>
</tr>
</tbody>
</table>

**Analogical reasoning**
### Phrase analogies

<table>
<thead>
<tr>
<th>Newspapers</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>New York Times</td>
<td>Baltimore</td>
<td>Baltimore Sun</td>
<td>Cincinnati Enquirer</td>
</tr>
<tr>
<td>San Jose</td>
<td>San Jose Mercury News</td>
<td>Cincinnati</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NHL Teams</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston</td>
<td>Boston Bruins</td>
<td>Montreal</td>
<td>Montreal Canadiens</td>
<td></td>
</tr>
<tr>
<td>Phoenix</td>
<td>Phoenix Coyotes</td>
<td>Nashville</td>
<td>Nashville Predators</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NBA Teams</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Detroit</td>
<td>Detroit Pistons</td>
<td>Toronto</td>
<td>Toronto Raptors</td>
<td></td>
</tr>
<tr>
<td>Oakland</td>
<td>Golden State Warriors</td>
<td>Memphis</td>
<td>Memphis Grizzlies</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Airlines</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Austria</td>
<td>Austrian Airlines</td>
<td>Spain</td>
<td>Spainair</td>
<td></td>
</tr>
<tr>
<td>Belgium</td>
<td>Brussels Airlines</td>
<td>Greece</td>
<td>Aegean Airlines</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Company executives</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Steve Ballmer</td>
<td>Microsoft</td>
<td>Larry Page</td>
<td>Google</td>
<td></td>
</tr>
<tr>
<td>Samuel J. Palmisano</td>
<td>IBM</td>
<td>Werner Vogels</td>
<td>Amazon</td>
<td></td>
</tr>
</tbody>
</table>
word2vec
Mikolov et al. (2013)

<table>
<thead>
<tr>
<th>Czech + currency</th>
<th>Vietnam + capital</th>
<th>German + airlines</th>
<th>Russian + river</th>
<th>French + actress</th>
</tr>
</thead>
<tbody>
<tr>
<td>koruna</td>
<td>Hanoi</td>
<td>airline Lufthansa</td>
<td>Moscow</td>
<td>Juliette Binoche</td>
</tr>
<tr>
<td>Check crown</td>
<td>Ho Chi Minh City</td>
<td>carrier Lufthansa</td>
<td>Volga River</td>
<td>Vanessa Paradis</td>
</tr>
<tr>
<td>Polish zolty</td>
<td>Viet Nam</td>
<td>flag carrier Lufthansa</td>
<td>upriver Russia</td>
<td>Charlotte Gainsbourg</td>
</tr>
<tr>
<td>CTK</td>
<td>Vietnamese</td>
<td>Lufthansa</td>
<td>Russia</td>
<td>Cecile De</td>
</tr>
</tbody>
</table>

Additive compositionality
Skip-gram with negative sampling increases $u_j^T h_t$ for real word-context pairs $\langle w_t, w_j \rangle$ and decreases it for noise pairs.

Given:
- a matrix of $d$-dim word vectors $W (|V_w| \times d)$
- a matrix of $d$-dim context vectors $U (|V_u| \times d)$

Skip-gram is implicitly factorizing the matrix $M = WU^T$.

What is $M$?
- Word-context matrix where each cell $(i, j)$ contains $PMI(w_i, w_j)$
- If number of negative samples $K > 1$, this is shifted by a constant $- \log K$
- (Assuming large enough $d$ and iterations)
Beyond words

Can we add word vectors to make sentence/paragraph/doc vectors?

\[ \text{doc } A = a_1 + a_2 + a_3 \]
\[ \text{doc } B = b_1 + b_2 + b_3 \]

\[
\cos(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \\
= \frac{1}{\|A\| \cdot \|B\|}(a_1 \cdot b_1 + a_1 \cdot b_2 + a_1 \cdot b_3 + a_2 \cdot b_1 + a_2 \cdot b_2 + a_2 \cdot b_3 + a_3 \cdot b_1 + a_3 \cdot b_2 + a_3 \cdot b_3)
\]

= weighted all-pairs similarity over \( A \) and \( B \)
Paragraph vector (a.k.a doc2vec)

Le & Mikolov (2014)

Distributed memory
- Predict target $w_t$ given context $w_{t-c}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+c}$ and doc label $d_k$
- At test time, hold $U$, $W$ fixed and back-prop into expanded $D$

$$
\ell(W, U) = -\log p(w_t|w_{t-c} \cdots w_{t+c}, d_k)
$$

Diagram:
- $w_t$
- $\ell(W, U)$
- $d_k$
- Input
- $D$
- $U$
- Softmax
- Label
- Loss
- Projection (concatenated)
Paragraph vector (a.k.a doc2vec)

Distributed Bag-of-Words (DBOW)

- Predict target n-grams $w_t, \ldots, w_{t+c}$ given doc label $d_k$
- At test time, hold $U$ fixed and back-prop into expanded $D$

\[
\ell(W, U) = - \sum_{i=0}^{c} \log p(w_{t+i}|d_k)
\]
Semantics are elusive

Visitors saw her duck with binoculars.

Did she duck or does she have a duck?

Who has the binoculars?

How many pairs of binoculars are there?
Semantics are elusive

Visitors saw her duck with binoculars.

Did she duck or does she have a duck?

Who has the binoculars?

How many pairs of binoculars are there?
Semantics are elusive

Visitors saw her duck with binoculars.

Did she duck or does she have a duck?

Who has the binoculars?

How many pairs of binoculars are there?
Visitors saw her duck with binoculars.

Did she duck or does she *have* a duck?

Who has the binoculars?

How many pairs of binoculars are there?
Semantics are elusive

Visitors saw her duck with binoculars.

Did she duck or does she have a duck?

Who has the binoculars?

How many pairs of binoculars are there?
Semantics are elusive

Did she duck or does she *have* a duck?

Who has the binoculars?

How many pairs of binoculars are there?
Semantics are elusive

Visitors saw her duck with binoculars.

Did she duck or does she have a duck?

Who has the binoculars?

How many pairs of binoculars are there?
Semantics are elusive

Visitors saw her duck with binoculars.

Did she duck or does she have a duck?

Who has the binoculars?

How many pairs of binoculars are there?
Semantics are elusive

Did she duck or does she *have* a duck?

Who has the binoculars?

How many pairs of binoculars are there?
Semantics are elusive

Visitors saw her duck with binoculars.

Did she duck or does she have a duck?

Who has the binoculars?

How many pairs of binoculars are there?
Task-specific representations

Many NLP tasks fall into classification or sequence tagging

Classification
- Given variable-length text $w_1 \cdots w_n$ (sentence, document, etc), find label $y$
- Normal discriminative approach:
  - Extract features over the input text
  - Train a linear classifier

Tagging
- Given variable-length text $w_1 \cdots w_n$, label spans $z_1, \ldots, z_m$
- Normal discriminative approach:
  - Distribute labels over input words (e.g., BIO, BILOU encodings)
  - Extract features over input words
  - Train a linear-chain conditional random field
Convolutional NNs

SENNNA
- Part-of-speech tagging
- Chunking
- Named entity recognition
- Semantic role labeling

Figure 1: Window approach network.

Figure 2: Sentence approach network.
Convolutional NNs

- Max-over-time pooling
- Two input embedding “channels” — one updated during training
Convolutional NNs

R-CNNs

- Capture interactions between non-consecutive \( n \)-grams
  e.g., not bad, not so bad, not nearly as bad

- Each \( n \)-gram context vector includes a weighted average over prior \( n \)-gram states

\[
c_{ij} = U^\top (W_1 x_i \odot W_2 x_j)
\]

\[
c_t = \sum_{i<t} \lambda_t c_{it}
\]

\[
h_t = \tanh(c_t + b)
\]

- Exponential computations avoided by dynamic programming

- Averaging weight learned as a recurrent gate

\[
\lambda_t = \sigma(W_\lambda x_t + U_\lambda h_{t-1} + b)
\]
Recurrent NNs

- Flexible models for classification and/or tagging
- Typically used with LSTM units or GRUs
- Bidirectional RNNs better for long inputs
Recurrent NNs

- Flexible models for classification and/or tagging
- Typically used with LSTM units or GRUs
- Bidirectional RNNs better for long inputs
Recurrent NNs

- Flexible models for classification and/or tagging
- Typically used with LSTM units or GRUs
- Bidirectional RNNs better for long inputs
Recursive NNs (a.k.a Tree-RNNs)

- Computation graph follows dependency or constituent parse
- Child-sum:
  - Good for arbitrary fan-out or unordered children
  - Suited to dependency trees (input $x_i$ is head word)
- $N$-ary:
  - Fixed number of children, each parameterized separately
  - Suited to binarized constituency parses (leaves take word inputs $x_i$)
Recursive NNs (a.k.a Tree-RNNs) - Computation graph follows dependency or constituent parse
- Child-sum:
  - Good for arbitrary fan-out or unordered children
  - Suited to dependency trees (input $x_i$ is head word)
- $N$-ary:
  - Fixed number of children, each parameterized separately
  - Suited to binarized constituency parses (leaves take word inputs $x_i$)
Text generation

Probabilistic language modeling
- Distribution over sequences of words $p(w_1, \ldots, w_T)$ in a language
- Typically made tractable via conditional independence assumptions

$$p(w_1, \ldots, w_n) = \prod_{t=1}^{T} p(w_t | w_{t-1}, \ldots w_{t-n})$$

- $n$-gram counts estimated from large corpora
- Distributions smoothed to tolerate data sparsity, e.g., Laplace (add-one) smoothing, Kneser-Ney smoothing
- Evaluate on perplexity over held-out data

$$2 \frac{1}{N} \sum_{i=1}^{N} \log_2 p(w_1^{(i)} \ldots w_{T_{i}}^{(i)})$$

Discriminative language modeling
- Estimate n-gram probabilities with a discriminative model

$$p(w_t | w_{t-1}, \ldots w_1) \approx f(w_1, \ldots, w_t)$$
NNLM

Model $p(w_t|w_{t-1}, \ldots w_{t-n})$ with a feed-forward neural net
Model $p(w_t|w_{t-1}, \ldots w_{t-n})$ with an RNN

Or an ensemble of multiple RNNs, randomly initialized

<table>
<thead>
<tr>
<th>Model</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kneser-Ney 5-gram</td>
<td>141</td>
</tr>
<tr>
<td>Random forest [Xu 2005]</td>
<td>132</td>
</tr>
<tr>
<td>Structured LM [Filimonov 2009]</td>
<td>125</td>
</tr>
<tr>
<td>Feedforward NN LM</td>
<td>116</td>
</tr>
<tr>
<td>Syntactic NN LM [Emami 2004]</td>
<td>110</td>
</tr>
<tr>
<td>RNN trained by BP</td>
<td>113</td>
</tr>
<tr>
<td>RNN trained by BPTT</td>
<td>106</td>
</tr>
<tr>
<td>4x RNN trained by BPTT</td>
<td>98</td>
</tr>
</tbody>
</table>

Results on Penn Treebank corpus
Model $p(w_t|w_{t-1}, \ldots w_{t-n})$ with an RNN
Or an ensemble of multiple RNNs, randomly initialized

Comparison of single RNN vs RNN ensembles
Recent models with character-CNN inputs and softmax alternatives
### Nearest neighbors in character-CNN embedding space

<table>
<thead>
<tr>
<th>Word</th>
<th>Top-1</th>
<th>Top-2</th>
<th>Top-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCERDIBLE</td>
<td>INCERDIBLE</td>
<td>NONEDIBLE</td>
<td>EXTENDIBLE</td>
</tr>
<tr>
<td><a href="http://WWW.A.COM">WWW.A.COM</a></td>
<td><a href="http://WWW.AA.COM">WWW.AA.COM</a></td>
<td><a href="http://WWW.AAA.COM">WWW.AAA.COM</a></td>
<td><a href="http://WWW.CA.COM">WWW.CA.COM</a></td>
</tr>
<tr>
<td>7546</td>
<td>7646</td>
<td>7534</td>
<td>8566</td>
</tr>
<tr>
<td>TOWNHALL1</td>
<td>TOWNHALL</td>
<td>DJc2</td>
<td>MOODSWING360</td>
</tr>
<tr>
<td>KOMARSKI</td>
<td>KOHARSKI</td>
<td>KONARSKI</td>
<td>KOMANSKI</td>
</tr>
</tbody>
</table>