Natural Language Processing (CSE 517): Cotext Models (II)

Noah Smith
© 2016

University of Washington
nasmith@cs.washington.edu

January 25, 2016

Thanks to David Mimno for comments.
Three Kinds of Cotext

If we consider a word token at a particular position \( i \) in text to be the observed value of a random variable \( X_i \), what other random variables are predictive of/related to \( X_i \)?

1. the document containing \( i \) (a moderate-to-large collection of other words)
2. the words that occur within a small “window” around \( i \) (e.g., \( x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2} \), or maybe the sentence containing \( i \))
3. a sentence known to be a translation of the one containing \( i \)
Three Kinds of Cotext

If we consider a word token at a particular position $i$ in text to be the observed value of a random variable $X_i$, what other random variables are predictive of/related to $X_i$?

1. the document containing $i$ (a moderate-to-large collection of other words)
Three Kinds of Cotext

If we consider a word token at a particular position \( i \) in text to be the observed value of a random variable \( X_i \), what other random variables are predictive of/related to \( X_i \)?

1. the document containing \( i \) (a moderate-to-large collection of other words)

2. the words that occur within a small “window” around \( i \) (e.g., \( x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2} \), or maybe the sentence containing \( i \)
Three Kinds of Cotext

If we consider a word token at a particular position $i$ in text to be the observed value of a random variable $X_i$, what other random variables are predictive of/related to $X_i$?

1. the document containing $i$ (a moderate-to-large collection of other words)
2. the words that occur within a small “window” around $i$ (e.g., $x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}$, or maybe the sentence containing $i$)
3. a sentence known to be a translation of the one containing $i$
Three Kinds of Cotext

If we consider a word token at a particular position $i$ in text to be the observed value of a random variable $X_i$, what other random variables are predictive of/related to $X_i$?

1. the document containing $i$ (a moderate-to-large collection of other words) $\rightarrow$ topic models

2. the words that occur within a small “window” around $i$ (e.g., $x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}$, or maybe the sentence containing $i$) $\rightarrow$ distributional semantics

3. a sentence known to be a translation of the one containing $i$ $\rightarrow$ translation models
Local Contexts: Distributional Semantics

Within NLP, emphasis has shifted from topics to the relationship between $v \in \mathcal{V}$ and more local contexts.

For example: LSI/A, but replace documents with “nearby words.” This is a way to recover word vectors that capture distributional similarity.

These models are designed to “guess” a word at position $i$ given a word at a position in $[i - w, i - 1] \cup [i + 1, i + 2]$.

Sometimes such methods are used to “pre-train” word vectors used in other, richer models (like neural language models).
Word2vec
(Mikolov et al., 2013a,b)

Two models for word vectors designed to be computationally efficient.

- Continuous bag of words (CBOW): $p(v \mid c)$
  - Similar in spirit to the feedforward neural language model we saw last time (Bengio et al., 2003)

- Skip-gram: $p(c \mid v)$

It turns out these are closely related to matrix factorization as in LSI/A (Levy and Goldberg, 2014)!
Skip-Gram Model

\[ p(C = c \mid X = v) = \frac{1}{Z_v} \exp c^T v \]

- Two different vectors for each element of \( \mathcal{V} \): one when it is “v” (\( v \)) and one when it is “c” (\( c \)).
- Like the log-bilinear model we saw last time, normalization term \( Z_v \) is expensive, so approximations are required for efficiency.
- Can expand this to be over the whole sentence or document, or otherwise choose which words “count” as context.
Word Vector Evaluations

See [http://wordvectors.org](http://wordvectors.org) for a suite of examples.

Several popular methods for *intrinsic* evaluations:

▶ Do (cosine) similarities of pairs of words' vectors correlate with judgments of similarity by humans?

▶ TOEFL-like synonym tests, e.g., rug → {sofa, ottoman, carpet, hallway}

▶ Syntactic analogies, e.g., "walking is to walked as eating is to what?" Solved via:

\[
\min_{v \in V} \cos (v_{\text{walking}} - v_{\text{walked}} + v_{\text{eating}})
\]
Word Vector Evaluations

See http://wordvectors.org for a suite of examples.

Several popular methods for *intrinsic* evaluations:

- Do (cosine) similarities of pairs of words’ vectors correlate with judgments of similarity by humans?

  
  ▶ TOEFL-like synonym tests, e.g., `rug` → `{sofa, ottoman, carpet, hallway}`

  ▶ Syntactic analogies, e.g., "walking is to walked as eating is to what?" Solved via:

    \[
    \min_{v \in V} \cos(v_w, v_{\text{walking}} - v_{\text{walked}} + v_{\text{eating}})
    \]
Word Vector Evaluations

See http://wordvectors.org for a suite of examples.

Several popular methods for \textit{intrinsic} evaluations:

- Do (cosine) similarities of pairs of words’ vectors correlate with judgments of similarity by humans?
- TOEFL-like synonym tests, e.g., \textit{rug} $\rightarrow$ $\{\text{sofa, ottoman, carpet, hallway}\}$
Word Vector Evaluations

See http://wordvectors.org for a suite of examples.

Several popular methods for *intrinsic* evaluations:

- Do (cosine) similarities of pairs of words’ vectors correlate with judgments of similarity by humans?
- TOEFL-like synonym tests, e.g., *rug* → \{*sofa, ottoman, carpet, hallway*\}
- Syntactic analogies, e.g., “*walking* is to *walked* as *eating* is to what?” Solved via:

\[
\min_{v \in V} \cos \left( v, v_{\text{walking}} - v_{\text{walked}} + v_{\text{eating}} \right)
\]
Word Vector Evaluations

See http://wordvectors.org for a suite of examples.

Several popular methods for *intrinsic* evaluations:

- Do (cosine) similarities of pairs of words’ vectors correlate with judgments of similarity by humans?
- TOEFL-like synonym tests, e.g., *rug* $\rightarrow \{\text{sofa, ottoman, carpet, hallway}\}$
- Syntactic analogies, e.g., “*walking* is to *walked* as *eating* is to what?” Solved via:

$$\min_{v \in V} \cos (v_v, v_{\text{walking}} - v_{\text{walked}} + v_{\text{eating}})$$

Also: *extrinsic* evaluations on NLP tasks that can use word vectors (e.g., sentiment analysis).
Recall the class-based bigram model:

\[
p(x_i \mid x_{i-1}) = p(x_i \mid z_i) \cdot p(z_i \mid z_{i-1})
\]

\[
= \theta_{x_i \mid z_i} \cdot \gamma_{z_i \mid z_{i-1}}
\]

\[
p(x, z) = \pi_{z_0} \prod_{i=1}^{\ell} \theta_{x_i \mid z_i} \cdot \gamma_{z_i \mid z_{i-1}}
\]

This is like a topic model where topic distributions are **bigram** distributed!

If we treat each \(z\) as latent—like in a topic model—we get to something very famous, called the **hidden Markov model** (HMM).
Comparing Five Models

- **unigram**
  - $x_1$, $x_2$, $x_3$, $x_4$

- **bigram (Markov)**
  - $x_1$, $x_2$, $x_3$, $x_4$

- **class-based bigram**
  - $z_1$, $z_2$, $z_3$, $z_4$

- **PLSA and LDA (topics)**
  - $z_1$, $z_2$, $z_3$, $z_4$

- **hidden Markov model**
  - $x_1$, $x_2$, $x_3$, $x_4$
Brown Clustering

There is a whole lot more to say about HMMs, which we’ll save for later.

Brown et al. (1992) focused on the case where each $v \in V$ is constrained to belong to only one cluster, $\text{cl}(v)$.

They developed a greedy way to cluster words hierarchically.
Brown Clustering: Sketch of the Algorithm

Given: $k$ (the desired number of clusters)

- Initially, every word $v$ belongs to its own cluster.
- Repeat $V - k$ times:
  - Find the pairwise merge that gives the greatest value for $p(x_{1:n}, z_{1:n})$.

It turns out this is equivalent to PMI for adjacent cluster values!

This is very expensive; Brown et al. (1992) and others (later) introduced tricks for efficiency. See Liang (2005) and Stratos et al. (2014), for example.
If you keep track of every merge, you have a *hierarchical* clustering.

Each cluster is a binary tree with words at the leaves and internal nodes corresponding to merges.

Indexing the merge-pairs by 0 and 1 gives a bit-string for each word; prefixes of each word’s bit string correspond to the hierarchical clusters it belongs to.

These can be seen as word embedings!
Brown Clusters from 56,000,000 Tweets

http://www.cs.cmu.edu/~ark/TweetNLP/cluster_viewer.html
Three Kinds of Context

If we consider a word token at a particular position \( i \) in text to be the observed value of a random variable \( X_i \), what other random variables are predictive of/related to \( X_i \)?

1. the document containing \( i \) (a moderate-to-large collection of other words) → topic models
2. the words that occur within a small “window” around \( i \) (e.g., \( x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2} \), or maybe the sentence containing \( i \)) → distributional semantics
3. a sentence known to be a translation of the one containing \( i \) → translation models
Let \( f \) and \( e \) be two sequences in \( \mathcal{V}^\dagger \) (French) and \( \mathcal{\bar{V}}^\dagger \) (English), respectively.

We’re going to define \( p(F \mid e) \), the probability over French translations of English sentence \( e \).

In a noisy channel machine translation system, we could use this together with source/language model \( p(e) \) to “decode” \( f \) into an English translation.

Where does the data to estimate this come from?
Let $\ell$ and $m$ be the (known) lengths of $e$ and $f$.

Latent variable $\mathbf{a} = \langle a_1, \ldots, a_m \rangle$, each $a_i$ ranging over $\{0, \ldots, \ell\}$ (positions in $e$).

- E.g., $a_4 = 3$ means that $f_4$ is “aligned” to $e_3$.

\[
p(f \mid e, m) = \sum_{\mathbf{a} \in \{0, \ldots, n\}^m} p(f, \mathbf{a} \mid e, m)
\]

\[
p(f, \mathbf{a} \mid e, m) = \prod_{i=1}^{m} p(a_i \mid i, \ell, m) \cdot p(f_i \mid e_{a_i})
\]

\[
= \delta_{a_i \mid i, \ell, m} \cdot \theta_{f_i \mid e_{a_i}}
\]
IBM Model 2, Depicted

PLSA and LDA (topics)

hidden Markov model

IBM 2
Parameter Estimation

Use EM!

E step: calculate posteriors over all $a_i$, and then soft counts (left as an exercise: what soft counts do you need?)

M step: use relative frequency estimation from soft counts to get $\delta$ and $\theta$
Variations

- IBM Model 1 is the same, but fixes $\delta_{j|i,\ell,m} = \frac{1}{\ell+1}$.
  - Log-likelihood is convex!
  - Often used to initialize IBM Model 2.
- Dyer et al. (2013) introduced a new parameterization:
  $$\delta_{j|i,\ell,m} \propto \exp(-\lambda \left| \frac{i}{m} - \frac{j}{\ell} \right|)$$
  (This is called fast_align.)
- IBM Models 3–5 (Brown et al., 1993) introduced increasingly more powerful ideas, such as “fertility” and “distortion.”
Wow! That was a lot of models!

We covered:

▶ Topic models: LSI/A, PLSA, LDA
▶ Distributional semantics models: Skip-gram, Brown clustering
▶ Translation models: IBM 1 and 2

All of them are probabilistic models that capture patterns of cooccurrence between words and context.

They do not have: morphology (word-guts), syntax (sentence structure), or translation dictionaries . . .
Readings and Reminders

- Collins (2011)
- Submit a suggestion for an exam question by Friday at 5pm.
- Form your project team by Wednesday 1/27.
- Project details will be finalized this week.


