Semantic space models for word meaning in context

Katrin Erk
University of Texas at Austin
Vector space models

- Represent a word through a vector of counts of other words observed in 20-word context window


- Similarity between targets modeled as distance in space
Applications of vector space models

- In natural language processing:
  - Information retrieval (Manning/Raghavan/Schütze 2008)
  - Ontology induction (Lin 98, Ravichandran et al 2005, Snow/Jurafsky/Ng 2006)
  - Word sense discrimination (Schütze 1998)

- In cognitive science:
  - Synonymy (Landauer&Dumais 1997, Pado&Lapata 2007)
  - Lexical priming (Lowe&McDonald 2000)
  - Similarity judgments (McDonald&Ramscar 2001)
Variants of vector space models (following Lowe 2001)

- Target items and dimensions (base elements):
  - Words / words
  - Words / documents or paragraphs
  - Documents or paragraphs / words
  - Words / pieces of syntactic context

- Further parameters:
  - Transformation about raw counts, e.g. inverse doc. frequency
  - Similarity measure
  - Dimensionality reduction on whole space
This talk: word meaning in context

- Focus of this talk:
  - Using vector spaces to model word meaning in context

- Modeling polysemy:
  - Bats flew out of the cave vs. He hit the ball with a bat.

- Vector space models: typically one vector per word
  - word type vector

- How to get to one vector per word in context?
  - word token vector
OVERVIEW

- Vector space models
- **Representing word meaning**
- Vector space models for word meaning in context
- The Structured Vector Space model
- Inferences over text: Textual Entailment
- Measuring applicability of inference rules
Modeling word meaning in context

- **Current main model: dictionary word senses**
  - Distinct senses
  - Assign single best sense for each occurrence of a word

- **Assigning best fitting word sense:**
  Hard task, for human annotator as well as systems

- **Does each word have a fixed list of distinct senses?**
  - Kilgarriff 2006, similarly Hanks 2000: Word uses often fall between dictionary definitions
  - Kintsch 2007: word meaning is “fluid and flexible”
    No clear best number of dictionary senses for a word
When is sense assignment difficult?

Example (by Diana McCarthy):

**match** has 9 senses in WordNet, including:

- 1. lighter consisting of a thin piece of wood or cardboard tipped with combustible chemical
- 3. burning piece of wood or cardboard
- 6. person regarded as a good matrimonial prospect
- 8. pair of people who live together
- 9. something that resembles or harmonizes with
The “match” example continued

- Residents say militants in a station wagon pulled up, doused the building in gasoline, and struck a **match**.
  - lighter

- This is at least 26 weeks by the week in which the approved **match** with the child is made.
  - sense 9: something that resembles or harmonizes with? “that tie makes a good match with your jacket”
  - sense 8: a pair of people who live together?
What do we see in the “match” example?

- “harmonize with”, “two people living together”: related senses
- Use similar to those senses, neither fits really well
- Similarly: Hanks (2000) on “bank”
  - Two clear senses: financial institution versus riverside
  - But how about “data bank”, “blood bank”
  - Clearly related to financial sense, but somewhat different.
Modeling word meaning: The challenge

- Traditional word sense disambiguation models word meaning through distinct senses
- But: Often difficult even for people to decide on the single best sense for an occurrence
  - Seems to happen with any dictionary
- Often difficult to decide how many senses a word should have
  - Hanks: “bank” example
Modeling word meaning without senses

- Hypothesis: Word meaning better modeled without list of distinct senses
  - Just look at different occurrences of a word
  - Graded notion of similarity between occurrences

- Some first support: annotation study by Erk, McCarthy & Gaylord 2009
  - Nuanced judgments on 5-point scale on word meaning in context
  - Seemed not to approximate coarse-grained senses

- But can graded representations also work better in applications?

- Occurrence similarity: natural to model in vector space
OVERVIEW

- Vector space models
- Representing word meaning
- **Vector space models for word meaning in context**
- The Structured Vector Space model
- Inferences over text: Textual Entailment
- Measuring applicability of inference rules
Vector space models of word meaning in context

- One word token vector per occurrence
  - Describes meaning in context without reference to dictionary senses
- Can observe degree to which token vectors group/cluster into distinct senses
  - May differ for different lemmas
  - bat.n versus show.v
Vector space models of word meaning in context

- Vector space models: typically one vector per word
  - word type vector

- How to get to one vector per word in context?
  - word token vector

- All token vector models:
  - Start with type vectors
  - Modify or combine
  - So, parametrized by underlying type vector model
Vector space models of word meaning in context: direct combination

- **Combine word type vectors to form word token vector**

- **Schütze 1998, Landauer&Dumais 1997:**
  - Combine all type vectors in the sentence context
  - Combine by summing or averaging vectors

- **Mitchell&Lapata 2008:**
  - Disambiguate verb by combining verb’s type vector with type vector of its subject
  - Combine by doing component-wise multiplication
Vector space models for word meaning in context: through neighbors

- **Select neighbor type vectors in space**

- **Kintsch 2001**
  - Disambiguate verb $V$ through one argument $N$
  - Neural network
  - $N$ determines activation among $V$’s 100 closest neighbors in vector space

- **Pantel and Lin (2002)**
  - Identify neighbor groups as senses
  - Static while Kintsch is dynamic
Vector space models for word meaning in context: encoding syntax

- **Encode syntactic structure into vector representation**

  - Smolensky 1990, Clark/Coecke/Sadrzadeh 2008:
    - Tensor product
      - Vector dimensionality grows as syntactic structure grows

  - Clarke 2008:
    - Infinite-dimensionality vector
The role of syntax

- **Take syntax into account?**
  - Yes, consider:
    - *Dog bites mailman* vs *Mailman bites dog*
  - Need to describe “who does what to whom”

- **Encode syntax into vector representation?**
  - No need
  - Alternative: Combine token vectors modularly with syntactic representation

Word token vectors used analogously to word senses
The role of syntax

- Modularly combining syntactic structure and word token vectors:
  - Need separate meaning representation for each word in the sentence
  - Not given in Schütze model, nor in Mitchell&Lapata model
    - Direct combination of vec(acquire), vec(skill) gives you same “meaning” for “acquire” and “skill”
  - Possible with Kintsch model
The role of syntax

- Syntax needs to influence computation of token vector
  - a horse *draws*... : pull
  - *draw* a horse : sketch
  - No existing approach to token vectors does this
OVERVIEW

- Vector space models
- Representing word meaning
- Vector space models for word meaning in context
- The Structured Vector Space model
- Inferences over text: Textual Entailment
- Measuring applicability of inference rules
The structured vector space (SVS) model (Erk & Pado 2008)

- “He caught the ball.”
  - Meaning of ‘ball’ in context of ‘catch’:
    Things that are typically caught
  - Meaning of ‘catch’ in context of ‘ball’:
    Typical actions performed with a ball

- Use representation of selectional preference for disambiguation
  - Characterization of typical arguments of a predicate
  - Typical arguments for the direct object position of “eat”: foodstuff
The structured vector space (SVS) model

- Include selectional preference (one vector per argument position) in representation of a word
- Also: “inverse selectional preference”
The structured vector space (SVS) model

- Selectional preference vector: Centroid (sum) of type vectors for typical argument headwords collected from a corpus
Computing meaning in context in SVS

Combine noun vector with selectional preference vector of verb. Combine verb vector with inverse sel. preference vector of noun.
SVS: properties

- Computing meaning in context:
  - One meaning-in-context vector per word,
  - Can be combined with syntactic representation

- Using selectional preferences:
  - Nouns disambiguated by selectional preferences of verbs
  - Verbs disambiguated by inverse sel. preferences of nouns

- Relation between words influences disambiguation:
  - different argument positions in general have different selectional preferences
  - “draw horse” versus “horse draws”
Evaluating models of word meaning in context: Paraphrase applicability

- Given a phrase,
  - His shoulders slumped.
  - Prices have slumped in the last week.

- How good is a given paraphrase in this context?
  - slouched?
  - declined?

- Modeling:
  - Compute vector space representation for “slump” in this sentence context
  - Paraphrase applicability = similarity to type vector of “slouch”
Paraphrase applicability: Datasets

- **Mitchell & Lapata 2008**
  - Subject/verb pairs like “shoulder slumped”,
  - Human ratings on goodness of paraphrases for the verb
  - Task: predict human ratings

- **McCarthy & Navigli 2007**
  - Full sentences from English Internet Corpus
  - Annotators provided paraphrases for one target word per sentence
  - Targets: verbs, nouns, adjectives, adverbs
  - Task: Given all paraphrases for a target, predict the ones that work in the given sentence
Paraphrase applicability: evaluation on Mitchell/Lapata dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>word space: $\rho$</th>
<th>syn space: $\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>M&amp;L</td>
<td>0.20**</td>
<td>0.24**</td>
</tr>
<tr>
<td>SVS</td>
<td>0.27**</td>
<td>0.22**</td>
</tr>
<tr>
<td>Baseline: type vector</td>
<td>0.0</td>
<td>0.08**</td>
</tr>
<tr>
<td>Ceiling</td>
<td>0.4</td>
<td>0.4</td>
</tr>
</tbody>
</table>

- Correlation (Spearman’s rho) with human judgments
  - **: $p < 0.01$
  - SVS performs at same level as M&L
Paraphrase applicability: evaluation on McCarthy/Navigli dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>V-Subj</th>
<th>V-Obj</th>
<th>N-Obj</th>
</tr>
</thead>
<tbody>
<tr>
<td>M&amp;L</td>
<td>50.3</td>
<td>52.0</td>
<td>53.4</td>
</tr>
<tr>
<td>SVS</td>
<td>63.1</td>
<td>55.8</td>
<td>56.9</td>
</tr>
<tr>
<td>Baseline: type vector</td>
<td>47.9</td>
<td>47.4</td>
<td>49.6</td>
</tr>
</tbody>
</table>

- Out-of-ten precision (McCarthy&Navigli 07): Out of the top ten paraphrases proposed by the model for a given occurrence, how many are gold paraphrases?

- SVS significantly outperforms M&L at p=0.05
SVS summary

- SVS: a vector space model of word meaning in context
  - Separate disambiguated vector for each word
  - Can be combined modularly with syntactic representation
  - Cross-wise disambiguation using (inverse) sel. preferences

- Performance:
  - Predicting paraphrase applicability:
    - same level as state-of-the-art M&L model that does not take syntax into account
  - But we now get one vector per word
OVERVIEW

- Vector space models
- Representing word meaning
- Vector space models for word meaning in context
- The Structured Vector Space model
- Inferences over text: Textual Entailment
- Measuring applicability of inference rules
Inference over text: Textual entailment

- Given a pair of sentences: Does the second follow from the first?

<table>
<thead>
<tr>
<th>Given statement</th>
<th>Second statement</th>
<th>Entailment</th>
</tr>
</thead>
<tbody>
<tr>
<td>LaFarge was the one who taught Tiffany to make the Favrile glass, said auctioneer William Doyle who operates an auction house in Manhattan</td>
<td>William Doyle works for an auction house in Manhattan</td>
<td>TRUE</td>
</tr>
<tr>
<td>Los Angeles, hometown of the Lakers, hosted the basketball championship final last night.</td>
<td>The Lakers were born in Los Angeles</td>
<td>FALSE</td>
</tr>
<tr>
<td>In 1999 Ford bought the apartment in Manhattan that he shares with his new girlfriend, Calista Flockhart.</td>
<td>Ford lives in Manhattan</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

Note: This is not entailment!
Why Textual Entailment is cool

- A “meta-application”: sentence pairs come from
  - information extraction
  - automatic summarization
  - question answering
Why Textual Entailment is cool

- Needs inference rules
  - X shares apartment in LOC -> X lives in LOC
- Needs rule applicability tests
  - LOC, hometown of X -> X was born in LOC ONLY IF X is a person
- Needs sophisticated rules over structure of sentence
  - For example, deal with negation:
    “Not even Stalin, it can hardly be denied, was always completely wrong”
OVERVIEW

- Vector space models
- Representing word meaning
- Vector space models for word meaning in context
- The Structured Vector Space model
- Inferences over text: Textual Entailment
- Measuring applicability of inference rules
Vector space representations and applicability of inference rules

- Applicability of inference rules central in Textual Entailment
- One important class of inference rules: paraphrase rules
  - X slump \(\rightarrow\) X decline
- Paraphrase rule applicability
  main evaluation measure for vector space models of word meaning in context
Inference rule applicability somewhat more generally

- Suggestion:
  An inference rule has one or more attachment points in space

- Occurrence close to attachment points triggers application of inference rule
Existing approaches described through attachment points

- Paraphrase applicability, above:
  - Rules like slump -> decline
  - Attachment point is type vector of rule RHS
  - Occurrence: token vector of rule LHS
Existing approaches described through attachment points

- Szpektor et al (2008): relation extraction
  - Rules like $X$ acquire $Y$ -> $X$ buy $Y$
  - First attachment point:
    - sum of type vectors in rule LHS and RHS
      - Compare to centroid of sentence
  - Further attachment points: selectional preferences for $X,Y$
    - Compare to arguments of “acquire” in sentence
Existing approaches described through attachment points

- Erk & McCarthy (2009): graded sense assignment
  - Given WordNet sense, locate it in space:
    - Occurrences of that sense in annotated corpus, e.g. SemCor
    - Then compute vector for sense as you would for a word

- Given WordNet sense, construct inference rules
  - Senses described through synonyms
  - Sense acquire.6 has synset {acquire, learn, larn}
  - Yields rules acquire -> learn, learn -> acquire, …
  - Attach at vector for sense acquire.6
From attachment points to attachment regions (Erk 2009)

- **Compute attachment region of a rule as**
  - centered on type vector
  - size determined by token vectors

- **Supervised learning of extent of the region:**
  - Positive training data: tokens of the word in question
  - Negative training data: tokens of other words
  - Maximum entropy model

- **Application: hyponymy (IS-A) rules**
  
  horse -> animal  
  mosey -> move
Conclusion

- Modeling word meaning
  - Difficult task
  - Hypothesis: Better modeled with graded representation than distinct senses (to be verified)
- SVS: a vector space model of word meaning in context
- Textual entailment: inferring one sentence from another
  - Needs inference rule applicability tests
- Modeling rule applicability through attachment points in space
Outlook

- Extend SVS: Model influence of wider context
- Evaluate on existing Textual Entailment datasets
- Attachment points for rules:
  - Much room for experimentation
- Integrate vector-based representation of word meaning with representation of sentence through logic
  - Deal with negation etc
  - Model gradedness in word meaning