Outline

- Introduction
- Vector Space Model
- Semantic spaces
- Embedded systems

Semantic spaces

- Ways of organizing the semantic entities
  - Distributional Semantics
    - Vectors, matrices, tensors
    - Different representations depending on POS
  - Compositional Semantics
    - Atomic units
      - Lexical semantics
    - Complex units
    - Relations between units
      - Ways of composition

Semantic spaces

– Semantic models of word co-occurrence

- Mitchell and Lapata, 2010; Grefenstette and Sadrzadeh, 2011 use a context window of 5 words on either side of the target word, and M = 2,000 vector dimensions.
- Model learned from BNC

$$BNC = \{Sen_1^{(BNC)}, \ldots, Sen_{n_{BNC}}^{(BNC)}\} \qquad Sen_i = (w_1^{(i)}, \ldots, w_{n_i}^{(i)})$$

- Base: M most frequent non stop words from $\{w_1^{(top)}, \ldots, w_M^{(top)}\}$

$$coCount_w[j] = \sum_{i=1}^{n_{BNC}} \sum_{t=1}^{n_i}$$

$$|\{k \in [t-5; t+5] \,|\, w_t^{(i)} = w,\; w_k^{(i)} = w_j^{(top)}\}|$$

Semantic spaces

– Semantic models of word co-occurrence

- Using these counts, word vectors are defined component-wise.

$$wdVec_w^{(rp)}[j] = \frac{p(w_j^{(top)}|w)}{p(w_j^{(top)})} =$$

$$\frac{coCount_w[j]}{freq_w} \times \frac{totalCount}{freq_{w_j^{(top)}}}$$

Semantic spaces

– Composition models

- **p** is the composition of two vectors, **u** and **v**.
- **p** = f(**u**,**v**)
- f is the composition function
- Additive Models:
  - **p** = **Au** + **Bv**
  - **A, B** are matrices expressing the relative contribution of **u** and **v** to **p**.
- Multiplicative Models:
  - **p** = **Cuv**
  - **C** is 3-rank tensor which project the tensor product of **u** and **v** into the space of **p**.

Semantic spaces

– Additional constraints

- components i are independent
- f is symmetric
  - $p_i = u_i + v_i$
  - $p_i = u_i \cdot v_i$
- relaxing symmetry
  - $p_i = \alpha u_i + \beta v_i$
- relaxing Independence:
  - circular convolution
    - » $p_i = \sum_j u_i \cdot v_{i-j}$
- Model combination
  - $p_i = \alpha u_i + \beta v_i + \gamma u_i v_i$

Semantic spaces

– Problem

- **u** and **v** are assumed homogeneous (the representation does not depend on Pos)

Semantic spaces

– Composition models

- Blacoe, Lapata 2012
- phrase vectors
  - Two words: <A, N>, <N, N>, <tV, N>, …
    - » additive

$$phrVec_{(w_1,w_2)} = wdVec_{w_1} + wdVec_{w_2}$$

    - » multiplicative

$$phrVec_{(w_1,w_2)} = wdVec_{w_1} \odot wdVec_{w_2}$$

- Sentence vectors

$$senVec_i^{(+)}[j] = \sum_{\substack{k=1,\ldots,n_i \\ wdVec_{w_k}\text{ exists}}} wdVec_{w_k}[j]$$

Semantic spaces

– Going beyond words
  - Should all syntactic categories of words be represented as vectors, or are some categories, such as adjectives, different?
  - does semantic composition factorize according to a constituency parse tree?
  - Mitchell & Lapata, 2008
  - Nouns are vectors ($\vec{n}$) in a n-dimensional Noun space.
  - Adjectives are n x n Matrices (A), so that A$\vec{n}$ = $\overrightarrow{an}$

Semantic spaces

– **Dependency-based construction of Semantic Space Models**
  - Padó, Lapata, 2007
  - Vector space models of word co-occurrence.
  - Syntactic structure in general and argument structure in particular is a close reflection of lexical meaning.
  - Paths, sequences of dependency edges extracted from the dependency parse of a sentence. Length of the path, labels considered.
  - Follows Lowe 4 steps process.
  - Basis mapping maps a path into an element of B.
  - 14 dependency relations (from Malt parser): amod, comp1, conj, fc, gen, i, lex-mod, mod, nn, obj, pcomp-n, rel, s, and subj.

Semantic spaces

- **DependencyVectors 2.0**
  - Based on Padó, Lapata, 2007
  - ExtractBasisElements
  - ExtractSpace
  - LLT (log-likelyhood transformation)

Semantic spaces

- **Distributional Semantic Models** (DSMs), Compositionality approaches by Marco Baroni's group:
  - Words are combined with linear matrices dependendent on the POS
- **Distributional Memory** (DM), a generalized framework for distributional semantics
- T set of target elements to be represented
- B set of base elements representing the context
- Sometimes T = B
- Modeling co-occurrence: M matrix, $M_{|B| \times |T|}$
  - Unstructured vs structured DSM
    - Just neighbours, corpus-derived triples

Semantic spaces

– DSM
  - Third-order tensors for representing word-link-word tuples.
  - Different semantic spaces are then generated on demand through the independently motivated operation of tensor matricization, mapping the third-order tensor onto two-way matrices. The matricization of the tuple tensor produces both familiar spaces, similar to those commonly used for attributional or relational similarity, and other less known distributional spaces, which will yet prove useful for capturing some interesting semantic phenomena.

Semantic spaces

– DSM

- Let W_1 and W_2 (frequently $W_1 = W_2 = W$) be sets of strings representing content words, and L a set of strings representing syntagmatic co-occurrence links between words in a text. $T \subseteq W_1 \times L \times W_2$ is a set of corpus-derived tuples $t = < w_1, l, w_2 >$ such that w_1 co-occurs with w_2 and l represents the type of this co-occurrence relation.

Semantic spaces

– Extracting semantic spaces from weighted triple tensor

word by link–word ($W_1{\times}LW_2$): vectors are labeled with words w_1, and vector dimensions are labeled with tuples of type $\langle l, w_2 \rangle$;

word–word by link ($W_1W_2{\times}L$): vectors are labeled with tuples of type $\langle w_1, w_2 \rangle$, and vector dimensions are labeled with links l;

word–link by word ($W_1L{\times}W_2$): vectors are labeled with tuples of type $\langle w_1, l \rangle$, and vector dimensions are labeled with words w_2;

link by word–word ($L{\times}W_1W_2$): vectors are labeled with links l and vector dimensions are labeled with tuples of type $\langle w_1, w_2 \rangle$.

Semantic spaces

- DSM implementations
  - Sources
    - ukWaC (U. Bolonia), 1.915 billion tokens
    - BNC, 95 million tokens
    - English WP (2009), 820 million tokens
  - Process
    - Tokenization, Pos tagging, dependency parsing
  - W: 30,693 lemmas (20,410 N, 5,026 V, 5,257 A)
  - Downloads
    - http://clic.cimec.unitn.it/dm/
    - Tensor (4GB)
    - Software

Semantic spaces

- DepDM
  - Dependency parse relations
- LexDM
  - Lexical material connecting words
- TypeDM
  - Patterns inside LexDM links

Semantic spaces

– Tasks

- $W_1 \times LW_2$
  - Similarity judgment
  - Synonym detection
  - Noun categorization
  - Selectional preferences
- $W_1W_2 \times L$
  - Analogy
  - Relation classification
  - Qualia extraction
  - Commonsense concept description
- …

Semantic spaces

– Focusing on Pos-based representations

- Grefenstette et al, 2011
- The meaning of a sentence is obtained by applying the function corresponding to the grammatical structure of the sentence to the tensor product of the meanings of the words in the sentence.
- Based on type-logics, some words will have atomic types and some compound function types. The compound types live in a tensor space where the vectors are weighted sums (i.e. superpositions) of the pairs of bases from each space. Compound types are “applied” to their arguments by taking inner products (similarly with compositional “Montague” semantics)

Semantic spaces

– Focusing on Pos-based representations

- “cats eat fish”
  - cats, fish: n
  - eat: $n^r s n^l$
- $n(n^r s n^l)n \rightarrow 1s1 \rightarrow s$
- Similar to Combinatory Categorial Grammar
- Syntactic $\rightarrow$ Semantic categories
  - n $\rightarrow$ semantic space N
  - $n^r s n^l$ $\rightarrow$ semantic space $N \otimes S \otimes N$
- the tensor space $A \otimes B$ has as a basis the cartesian product of a basis of A with a basis of B.

Semantic spaces

- if $(\overrightarrow{v_1}, \overrightarrow{v_2} \ldots, \overrightarrow{v_n})$ is a basis of A, then any vector $\vec{a} \in$ A can be written as $\vec{a}$ = $\sum_i C_i \overrightarrow{v_i}$
- if $(\overrightarrow{v_1}, \overrightarrow{v_2} \ldots, \overrightarrow{v_n})$ is a basis of A and $(\overrightarrow{v'_1}, \overrightarrow{v'_2} \ldots, \overrightarrow{v'_n})$ is a basis of B
- a vector $\vec{c}$ in the tensor space A $\otimes$ B can be written as $\sum_{i,j} C_{ij} (\overrightarrow{v_i} \otimes \overrightarrow{v'_j})$

Semantic spaces

$$\begin{aligned}
\overrightarrow{\text{dogs chase cats}} &= (\epsilon_N \otimes 1_s \otimes \epsilon_N)\left(\overrightarrow{\text{dogs}} \otimes \overrightarrow{\text{chase}} \otimes \overrightarrow{\text{cats}}\right) \\
&= (\epsilon_N \otimes 1_s \otimes \epsilon_N)\left(\overrightarrow{\text{dogs}} \otimes \left(\sum_{ijk} C_{ijk}(\overrightarrow{n_i} \otimes \overrightarrow{s_j} \otimes \overrightarrow{n_k})\right) \otimes \overrightarrow{\text{cats}}\right) \\
&= \sum_{ijk} C_{ijk} \langle \overrightarrow{\text{dogs}} \mid \overrightarrow{n_i} \rangle \overrightarrow{s_j} \langle \overrightarrow{n_k} \mid \overrightarrow{\text{cats}} \rangle
\end{aligned}$$

Semantic spaces

- Adjective phrases
  - adjectives
    - » nn^l
    - » $N \otimes N$
    - » $nn^l n \rightarrow n1 \rightarrow n$

$$\overrightarrow{\text{red fox}} = (1_N \otimes \epsilon_N)(\overrightarrow{\text{red}} \otimes \overrightarrow{\text{fox}}) = \sum_{ij} C_{ij} \overrightarrow{n_i} \langle \overrightarrow{n_j} \mid \overrightarrow{\text{fox}} \rangle$$

Semantic spaces

- Learning the parameters of transitive verbs
  - Grefenstette & Sadrzadeh, 2011

$$\overrightarrow{\text{sub tverb obj}} = \underline{\text{tverb}} \odot (\overrightarrow{\text{sub}} \otimes \overrightarrow{\text{obj}})$$

  - $\overrightarrow{sub}$ and $\overrightarrow{obj}$ are vectors in the r-dimensión space N, so they can be learned directly through their occurrences in the corpus.
  - tverb is a r x r matrix. It can be learned:
    - » Indirectly
      - » constructed by taking the sum of the kronecker products of all of the subject/object pairs linked to that verb in the corpus.
    - » Directly
      - » Getting the $\overrightarrow{tverb}$ vector from the corpus
      - » Mapping this vector into a matrix

Semantic spaces

- Mapping the verb vector into a matrix
  - 0-diagonal

$$\textbf{0-diag}: \begin{pmatrix} c_1 s_1 o_1 & 0 & \cdots & 0 \\ 0 & c_2 s_2 o_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_r s_r o_r \end{pmatrix}$$

  - 1-diagonal

$$\textbf{1-diag}: \begin{pmatrix} c_1 s_1 o_1 & s_1 o_2 & \cdots & s_1 o_r \\ s_2 o_1 & c_2 s_2 o_2 & \cdots & s_2 o_r \\ \vdots & \vdots & \ddots & \vdots \\ s_r o_1 & s_r o_2 & \cdots & c_r s_r o_r \end{pmatrix}$$

  - tverb ⊗ tverb

$$v \otimes v: \begin{pmatrix} c_1 c_1 s_1 o_1 & c_1 c_2 s_1 o_2 & \cdots & c_1 c_r s_1 o_r \\ c_2 c_1 s_2 o_1 & c_2 c_2 s_2 o_2 & \cdots & c_2 c_r s_2 o_r \\ \vdots & \vdots & \ddots & \vdots \\ c_r c_1 s_r o_1 & c_r c_2 s_r o_2 & \cdots & c_r c_r s_r o_r \end{pmatrix}$$

Outline

- Introduction
- Vector Space Model
- Semantic spaces
- Embedded systems

Embeddings

– Distributional Semantics

  - most recent effort towards solving this problem concern latent factor models because they tend to scale better and to be more robust w.r.t. the heterogeneity of multi-relational data.
  - These models represent entities with latent factors (usually low-dimensional vectors or **embeddings**) and relationships as operators for combining those factors.
  - Operators and latent factors are trained to fit the data using reconstruction, clustering or ranking costs.:

Embeddings

- Embedding Methods for NLP
  - Weston & Bordes, EMNLP tutorial 2014
- Deep Learning
- Similar words should have similar embeddings (share latent features).
- Embeddings can also be applied to symbols as well as words (e.g. Freebase nodes and edges).
- Can also have embeddings of phrases, sentences, documents, or even other modalities such as images.

Embeddings

- Embedding Models
  - Models based on low-dimensional continuous vector embeddings for entities and relation types, directly trained to define a similarity criterion.
  - Stochastic training based on ranking loss with sub-sampling of unknown relations.

NN models for NLP

- This topic will be covered later in this course