Distributional Semantics and Compositionality

Mitchell&Lapata 2008, Erk&Pado 2008

Katerina Danae Kandylaki

Saarland University

May 10, 2010

Outline

Semantic Composition

Main question: how to compose the meaning of words into phrases/ sentences.

Examples

1. catch a ball
2. catch a disease
3. attend a ball

Composition:

1. "<u>catch</u>" + "<u>ball</u>" → "grab", "spherical object"
2. "<u>catch</u>" + "disease" → "contract"
3. "attend" + "<u>ball</u>" → "dancing event"

The importance of argument positions

What happens for cases like this??

Different argument positions

Example

- a horse draws
- draw a horse

Need for accounting the word's **selectional preferences** for its argument positions.

Vector-based Semantic Composition

In a vector based model of word meaning:

- **vector** : represents an individual word or a composition of words (depending on the model)
- **dimensions** : the possible co-occurrent words (semantically similar)
- **vector spaces** : built from corpora; use of vector spaces in the evaluation task

Semantic Composition: vector composition (in various ways)

How can we compose vectors?

Operations on vectors:

1. ***vector addition***: two vectors **v** and **w** can be "added" to yield the sum **v** + **w** (which is another vector)

2. ***multiplication***

Note: operations are performed component-wise

Existing approaches

Existing approaches:

- **General context effects** (Schütze 1998, McDonald&Brew 2004) "first-" and "second-order" vectors, result sense clusters.
- **Predicate-argument combination** (Kintsch 2001, Mitchell&Lapata 2008), context typically consists of a single word, no/little account for the relation between them.
- **Integration of lexical information and selectional preferences** (Erk&Pado 2008), within the framework of Mitchell&Lapata 2008.

Mitchell&Lapata 2008 (M&L model)

propose a framework for vector composition which:

- allows **the derivation of different types of models** and
- licenses two fundamental composition operations, **multiplication and addition** (and their combination).

Basic assumptions

- *Semantic composition* : a function of two vectors, u and v.
- **Individual words** : vectors acquired from a corpus following parametrisation.
- **word vector** : typically represents its co-occurrence with neighbouring words.

Semantic space construction

Construction of a semantic space:

- **definition of** linguistic **context** (e.g., neighbouring words can be documents or collocations)
- number of **components** used (e.g., the k most frequent words in a corpus)
- **values** of these components, and their values (e.g., as raw co-occurrence frequencies or ratios of probabilities).

Example

A hypothetical semantic space for *horse* and *run* (five-dimensional space, matrix cells: co-occurrence of the target words):

	animal	stable	village	gallop	jockey
horse	0	6	2	10	4
run	1	8	4	4	0

General class of models

General class of models:

$$\mathbf{p} = f(\mathbf{u}, \mathbf{v}, R, K)$$

- p: the composition of two vectors, u and v.
- u, v: vectors of the constituents that stand in some syntactic relation
- R: the syntactic relation
- K: additional knowledge or information needed to construct the semantics of the composition

Adding constraints; reduced class

1. Hold R **fixed**: focus on a single linguistic structure, e.g. verb-subject relation
2. **Ignore** K: explore what can be achieved in the absence of additional knowledge

Reduced class of models:

$$\mathbf{p} = f(\mathbf{u}, \mathbf{v})$$

Specifying the form of f

Addition

- **Additive** model (simple instantiation):

$$\mathbf{p_i} = \mathbf{u_i} + \mathbf{v_i}$$

Example

horse + **run** = [1 14 6 14 4]

	animal	stable	village	gallop	jockey
horse	0	6	2	10	4
run	1	8	4	4	0

Problem: all components have the same contribution to the composition

Specifying the form of f

Weighted Addition

Adding weights to vector components:

$$\mathbf{p_i} = \alpha \mathbf{u_i} + \beta \mathbf{v_i}$$

Weighting the contribution of the two components differently $\rightarrow$ **Syntax awareness**

Example

if α= 0.4, β= 0.6, we have:
horse = [0 2.4 0.8 4 1.6]
run = [0.6 4.8 2.4 2.4 0]
their sum **horse** + **run** = [0.6 7.2 3.2 6.4 1.6]

	animal	stable	village	gallop	jockey
horse	0	6	2	10	4
run	1	8	4	4	0

Specifying the form of f

Multiplication

- **Multiplication** model (simplified version):

$$\mathbf{p_i} = \mathbf{u_i} \cdot \mathbf{v_i}$$

Example

horse·**run** $= [0\ 48\ 8\ 40\ 0]$

	animal	stable	village	gallop	jockey
horse	0	6	2	10	4
run	1	8	4	4	0

Specifying the form of f

Multiplication - Components with value zero

Example

bank (selected senses): 1. depository financial institution, 2. sloping land

	park	cashier
bank	10	8
account	0	8

bank·account = [0 64]: component-wise multiplication cancels out the irrelevant meaning of bank in this specific context.

Specifying the form of f

Combination of additive and multiplicative methods

Combination of methods:

$$\mathbf{p_i} = \alpha\mathbf{u_i} + \beta\mathbf{v_i} + \gamma\mathbf{u_i}\mathbf{v_i}$$

Advantages:

- Account for syntax
- Migrates possible problems from zero components

Note: This is the model to be conpared with the Erk&Pado model in the Evaluation section.

Erk&Pado 2008 (SVS model)

present a robust model of semantic composition, the *structured vector space* model:

- *selectional preferences* for a word's argument positions
- syntactic information in the computation of a word meaning in context

Weak point of existing models; phrases

Remember the case:

- a horse draws
- draw a horse

"horse" (**Subj**) + "draw" : verb similar to "pull"

vs.

"draw" + "horse" (**Obj**) : verb similar to "sketch"

Different **syntactic relation** between components: not captured by a **single vector representation**.

Weak point of existing models; sentences

How can semantic spaces **“scale up”** to provide representations for entire sentences?

Consider the sentences:

- The dog chased the cat.
- The cat chased the dog.

Vectors: **fixed dimensions** -> **fixed amount** of structure representation

Sentences: **arbitrary** amount sentence length and structural information

Therefore, single vector representations are **not efficient** to encode an arbitrary amount of structures

Basic idea behind the SVS model

Main intuition: the interpretation of a word in context is guided by *expectations about typical events*.

Example

"catch a ball"

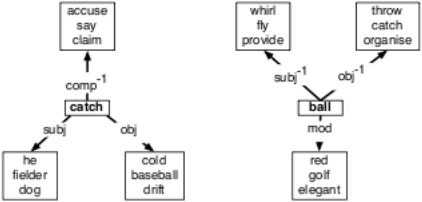
- "catch" matches with typical actions that can be performed with a ball
- "ball" reflects the hearer's expectations of typical things that can be caught

linguistic expectations = selectional preferences

SVS model: Representing lemma meaning

Word is encoded as a **combination** of:

1. one vector that models the *lexical meaning* of the word
2. a set of vectors, representing the *semantic expectations/ selectional preferences* for one particular relation that the word supports.



Formal description of a lemma

In a vector space D (the set of possible vectors), with R the set of relation labels; a lemma is represented by the triple:

$$w = (v, R, R^{-1})$$

- $v \in D$: lexical vector of the word w itself,
- $R : R \to D$: w's selectional preferences
- $R^{-1} : R \to D$: inverse selectional preferences of w.

Note: Both R and R^{-1} are partial functions.

Computing meaning in Context

From the lexicon, we have:

- word a: $a = (v_a, R_a, R_a^{-1})$
- word b: $b = (v_b, R_b, R_b^{-1})$
- $r \in R$: the relation linking a to b.

We **compose** the meaning of a and b as a pair (a', b') of vectors:

- a': the meaning of a in the context of b
- b': the meaning of b in the context of a

Formal description of the composition:

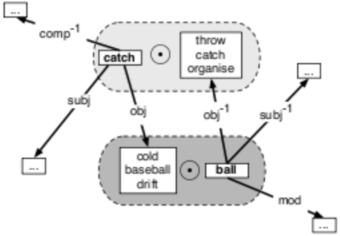
$$a' = (v_a \odot R_b^{-1}(r), R_a - \{r\}, R_a^{-1})$$

$$b' = (v_b \odot R_a(r), R_b, R_b^{-1} - \{r\})$$

Computing meaning in Context (2)

Example

"catch a ball"



$$a' = (v_a \odot R_b^{-1}(r), R_a - \{r\}, R_a^{-1})$$

$$b' = (v_b \odot R_a(r), R_b, R_b^{-1} - \{r\})$$

Result: one context-adapted meaning representation per word

Computation of elementary vectors

Computing elementary vectors in this case means constructing the vector space:

- **Bag Of Words (BOW)** vector space: for target- context word pair, consider their co-occurrence frequency within a surface window of size 10.
- **"Dependency-based" (SYN)** vector space : target and context words need to be linked by a "valid"[1] dependency relation to count as co-ocurring.

[1] "valid" relations were defined using minimal context specification and plain weight function

Computation of Selectional Preferences

Model selectional preferences through **similarity to seen filler vectors** v_a

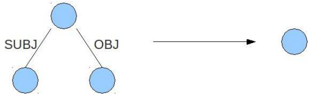
- Compute the selectional preference for word b and relation (between a and b) r :
  - SELPREF (baseline) model

$$R_b(r)_{SELPREF} = \sum_{a:f(a,r,b)>0} f(a,r,b) \cdot v_a$$

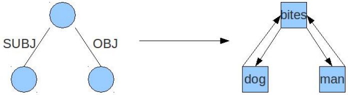
- Two additional variants alleviating noise:
  - SELPREF-CUT: instead of all fillers, the model uses only the fillers seen more often than a threshold θ.
  - SELPREF-POW: takes each component of the selectional preference vector to the nth power.

Schematic comparison of M&L vs. SVS model

M&L: **single vector representation**



SVS: **structured vector representation (one per word in context)**



Experiment 1

Human similarity judgements

- **Experiment 1**: one construction limit - verb and its subject
- **Task**: predict human similarity judgements

Example

human similarity judgements

verb	subject	landmark	sim	judgement
slump	shoulder	slouch	high	7
slump	shoulder	decline	low	2
slump	value	slouch	low	3
slump	value	decline	high	7

Experiment 1
Results

- Mean **cosine similarity** for items with high- and low-similarity landmarks

Model	high	low	ρ	high	low	ρ
	BOW	space		SYN	space	
Target only	0.32	0.32	0.0	0.20	0.20	0.08
SelPref only	0.46	0.40	0.06	0.27	0.21	0.16
M&L	0.25	0.15	**0.20**	0.13	0.06	**0.24**
SELPREF	0.32	0.26	0.12	0.22	0.16	0.13
SELPREF-CUT, θ=10	0.32	0.24	0.11	0.20	0.13	0.13
SELPREF-POW, n =30	0.11	0.03	**0.27**	0.08	0.04	**0.22**
Upper bound	-	-	0.4	-	-	0.4

Experiment 2

Appropriate vs. inappropriate paraphrases

- **Experiment 2**: more types of syntactic constructions
- **Task**: distinguish between appropriate and inappropriate paraphrases

Example

Lexical substitution items for "work"

Sentence	Substitutes
By asking people who work there, I have since determined that he didn't.	be employed 4; labour 1
Remember how hard your ancestors worked.	toil 4; labour 3; task 1

Experiment 2
Results

- Mean "out of ten" percision (P_{oor})

Model	V-SUBJ	V-OBJ	N-OBJ
Target only	47.9	47.4	49.6
SelPref only	54.8	51.4	55.0
M&L	50.3	52.0	53.4
SELPREF-POW, n =30	**63.1**	**55.8**	**56.9**

Encouraging evidence for the usefulness of selectional preferences for judging substitutability in context

Conclusions

Vector- based Semantic Composition:

1. M&L model:
   - word/ phrase/ sentence represented by a single vector
   - limited coverage of syntax
2. SVS model:
   - lemma: lexical information + selectional preferences
   - composition: combination of selectional preferences of words in context
   - account for syntactic relations, more efficient encoding of meaning

General conclusion: A word's selectional preferences play a central role in the computation of meaning in context.

Thank you!