Some challenges for Distributional Semantic Models

Stefan Evert[1], Alessandro Lenci[2]

[1]University of Osnabrück
[2]University of Pisa

Bordeaux, July 31 2009

Outline

Many tasks, many DSMs

- Many (computational) linguistic and cognitive tasks can be modeled with DSMs
  - synonym identification
  - semantic similarity judgment
  - categorization
  - analogy recognition
  - semantic relation classification
  - selectional preference modeling
  - argument alternations
  - nomina actionis recognition
  - ...
- Different tasks seem to require different distributional spaces
  - word X word
  - word pair X link
  - verb slot X filler
  - ...

"One task, one model"

The standard approach in corpus-based semantics

- For each semantic task...
  - taxonomic similarity, relation identification, selectional preferences, etc.
- ...develop a different corpus-based pipeline
- Excellent empirical results but:
  - not what humans do (human *semantic memory* is general-purpose)
  - computationally inefficient, resources rarely reusable, prone to overfitting

Towards a unified model

- Turney (2008)
  - various tasks are reinterpreted as instances of a more general task, i.e. analogy recognition
- Baroni & Lenci (2009)
  - *"One semantic memory, many semantic tasks"*
  - each task may keep its specificity
  - unification is achieved by designing a sufficiently general distributional structure, from which semantic spaces can be generated on demand

Distributional Memory

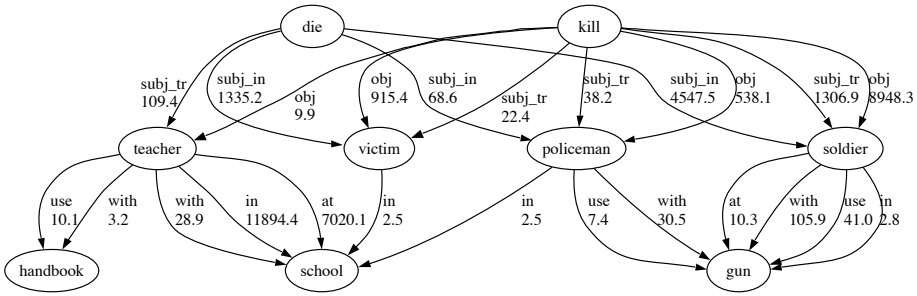
Baroni & Lenci (2009)

- DM: a graph of weighted links between concepts, built once and for all from the source corpus
  - *same* co-occurrence statistics parameters, *same* target words, *same* weighting scheme
- Different semantic tasks tackled by extracting different matrices from underlying DM graph

Trained DM

$concept_1$	link	$concept_2$	weight
soldier	use	gun	41.0
gun	use^{-1}	soldier	41.0
policeman	with	gun	30.5
gun	$with^{-1}$	policeman	30.5
kill	obj	victim	915.4
victim	obj^{-1}	kill	915.4
kill	subj_tr	soldier	1306.9
soldier	$subj\_tr^{-1}$	kill	1306.9

DM as a graph



Matrix views of the DM graph

CxLC

	$subj\_in^{-1}$ die	$subj\_tr^{-1}$ kill	obj^{-1} kill	with gun	use gun
teacher	109.4	0.0	9.9	0.0	0.0
soldier	4547.5	1306.9	8948.3	105.9	41.0
policeman	68.6	38.2	538.1	30.5	7.4

CCxL

		in	at	with	use
teacher	school	11894.4	7020.1	28.9	0.0
teacher	handbook	2.5	0.0	3.2	10.1
soldier	gun	2.8	10.3	105.9	41.0

CLxC

		teacher	victim	soldier	policeman
kill	subj_tr	0.0	22.4	1306.9	38.2
kill	obj	9.9	915.4	8948.3	538.1
die	subj_in	109.4	1335.2	4547.5	68.6

Training DM

- Training corpus
  - *ukWaC*, ukWaC, 2.25 billion tokens from the Web (Ferraresi et al, 2008), pre-parsed with MINIPAR (Lin 1998)
- Concepts
  - Top 20k most frequent nouns, top 5k most frequent verbs
- Links
  1. the top 30 most frequent direct V-N dependency paths (e.g. *kill+obj+victim*)
  2. the top 30 preposition-mediated N-N or V-N paths (e.g. *soldier+with+gun*)
  3. the top 50 transitive-verb-mediated N-N noun paths (e.g. *soldier+use+gun*)
  4. all the inverse relations of (1)-(3) (e.g. *victim+obj*$^{-1}$*+kill*)
- Weights
  - Local MI (Evert 2005)
- DM size
  - 69 million tuples

Semantic tasks taken by DM (so far. . .)

- Concept-by-Link+Concept (CxLC)
  1. semantic similarity judgments
  2. noun categorization
  3. verb selectional restrictions
- Concept+Concept-by-Link (CCxL)
  4. recognizing SAT analogies
  5. semantic relation classification
- Concept+Link-by-Concept (CLxC)
  6. argument alternations

- The emphasis is on the model generality and adaptivity
  - the goal is to achieve state-of-the-art results (*not necessarily the best score*), without task-specific parameter tuning

DM: a general framework for DSMs

Baroni & Lenci (2009)

- DM achieves state-of-the-art results in various tasks, without resorting to any "task-specific" optimization
- Many semantic tasks can be tackled by assuming that there is an underlying tuple-based "distributional memory"
- Different ways to build co-occurrence matrices from the distributional graph generate different semantic spaces
  - each space is a different "semantic view" on the underlying distributional graph

Outline

Compositionality

- Compositionality is a core aspect of natural language, and in particular natural language semantics

word	*type*	*logical form*	*meaning*
Tom	e	**Tom**	Tom, the cat
chases	$< e, < e, t >>$	$\lambda y \lambda x$.**chase**(x,y)	$\{< x, y > \mid$ x chases y$\}$
Jerry	e	**Jerry**	Jerry, the mouse

- *Tom chases Jerry* $\Rightarrow$ **chases**(**Tom**, **Jerry**)
  - TRUE iff in this world Tom the cat chases Jerry the mouse
- Compositionality allows us to create infinite meanings with finite means

The principle of compositionality

The principle of compositionality

The meaning of a complex expression is a function of the meanings of its parts and of their syntactic mode of combination

- The ingredients of compositionality (Partee 1984)
  - a theory of lexical meanings - assigns meanings to the smallest part (e.g. words)
  - a theory of syntactic structures - determines the relevant part-whole structure of each complex expression
  - a theory of semantic composition - determines the *combinatorial semantic operations*, i.e. the functions that compose the meanings

Outline

Words have multiple meanings

- A key problem for compositionality
  - each element to be composed must have a unique and constant meaning
- Trivially solved (or rather ignored...) in formal semantics
  - *The cat chases the mouse* $\Rightarrow$ **mouse**$_1$
  - *The hacker clicks the mouse button* $\Rightarrow$ **mouse**$_2$
- A potential problem for DSMs too
  - meaning is represented by vectors (i.e. matrix rows)
  - there is one vector per each word type, and this vector conflates its different senses
    - *The cat chases the mouse* $\Rightarrow \overrightarrow{mouse}$
    - *The hacker clicks the mouse button* $\Rightarrow \overrightarrow{mouse}$
  - $\overrightarrow{mouse}$ will include information about the typical contexts of *mouse-as-animal* and the typical contexts of *mouse-as-device* (with a bias towards the most frequent sense in a corpus)

Distinguishing word senses with DSMs

- Word senses are distinguished by contexts
- The context of a word token can also receive a representation in DSMs
- Context vectors (Schütze 1998)
  - for each word token w_i, take the words in its context C_i

    C_1 = {cat, chase}
    C_2 = {hacker, click, button}
  - for each C_i, build a context vector $\overrightarrow{C_i}$, by summing the vectors in a DSM of the words in C_i

    $\overrightarrow{C_1} = \overrightarrow{cat} + \overrightarrow{chase}$
    $\overrightarrow{C_2} = \overrightarrow{hacker} + \overrightarrow{click} + \overrightarrow{button}$
  - each context vector is the centroid of the vectors of its words

Distinguishing word senses with DSMs

Schütze 1998

- Word senses are represented by clusters of similar contexts
  - e.g. the cluster of the contexts of mouse-as-animal

1. take all the contexts of a word w in a training corpus
2. build the context vector $\overrightarrow{C_i}$, for each of these contexts
3. cluster the context vectors
4. for each cluster, takes the centroid vector of the cluster, and use this vector to represent one sense of w (sense vector, $\overrightarrow{s_j}$)

- To assign a sense to a new instance of w in context C_k

1. build the context vector $\overrightarrow{C_k}$
2. assign to w in context C_k the sense j whose sense vector $\overrightarrow{s_j}$ is closest to $\overrightarrow{C_k}$

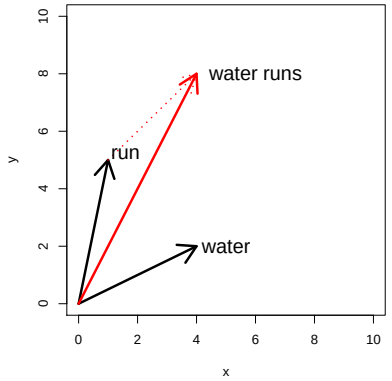
Outline

Semantic operations

- Let's assume that we have "solved" the problem of assigning a specific interpretation to each words in a complex expression
- The crucial step to account for compositionality is to identify the semantic operations to combine the meaning of atomic elements into a meaning for the complex expression
- The formal approach
  - the key operation for semantic composition is type-driven functional application (possibly integrated with other types of operations; cf. McNally's course at ESSLLI'09, Pustejovsky 1995, etc.)
    - $\lambda y \lambda x$.**chase**(x,y)(**Jerry**) = λx.**chase**(x,**Jerry**) $\Rightarrow$ $\{< x, Jerry > \mid$ x chases Jerry the mouse$\}$

Composition as vector combination

- Given two word vectors, the distributional "meaning" of the phrase combining them is given by a combined vector (e.g., the sum of their vectors)
  - Landauer & Dumais (1996), Kintsch (2001), Mitchell & Lapata (2008)



Types of vector composition

Mitchell & Lapata (2008)

- Simple vector sum (Landauer & Dumais 1996)
  - $\overrightarrow{p} = \overrightarrow{a} + \overrightarrow{b}$
    - $\overrightarrow{chase\ cat} = \overrightarrow{chase} + \overrightarrow{cat}$
- Contexts-sensitive vector sum (Kintsch 2001)
  - $\overrightarrow{p} = \overrightarrow{a} + \overrightarrow{b} + \sum \overrightarrow{n}$
    - n are the $n-$top nearest neighbors of the predicate
    - $\overrightarrow{chase\ cat} = \overrightarrow{chase} + \overrightarrow{cat} + (\overrightarrow{hunt} + \overrightarrow{prey} + \ldots + \overrightarrow{capture})$
  - Kintsch captures effects of context-sensitivity in predication (e.g. disambiguation, co-composition, metaphorical interpretation, etc.)
- Vector pairwise moltiplication (Mitchell & Lapata 2008)
  - $\overrightarrow{p} = \overrightarrow{a} \cdot \overrightarrow{b}$
    - $\overrightarrow{chase\ cat} = \overrightarrow{chase} \cdot \overrightarrow{cat}$

Additive vs. multiplicative composition

- Additive composition preserves all the dimensions of the component vectors

	hacker	cheese	button
mouse	25	10	17
click	30	0	20
click mouse	55	10	37

- Multiplicative composition selects only the dimensions shared by the component vectors

	hacker	cheese	button
mouse	25	10	17
click	30	0	20
click mouse	1650	0	340

- Mitchell & Lapata (2008) report better results with multiplicative methods (tested on a lexical paraphrase task)

Composition as vector combination

- It is useful to extract the general gist of a passage (cf. Landauer & Dumais 1996)
- However, order/hierarchical structure is not taken into account
  - *The cat chases the mouse* and *The mouse chases the cat* produce identical vectors
- The interpretation of a composed vector is not clear
  - $\overrightarrow{chase\ cat}$ is something in between $\overrightarrow{chase}$ and $\overrightarrow{cat}$, but the meaning of *chase cat* is not something in between the meaning of *chase* and the meaning of *cat*
- With vector multplication we can select specific senses of a word in context, but often multiple senses are simultaneously active (cf. co-predication)
  - *John leaves in a large city in the north of the US*
  - *This city voted for Obama*
  - *Large cities in the north of the US voted for Obama*

Making vector combination more sophisticated

- Jones and Mewhort (2007), following up on work by Smolensky (1990) and Plate (2003), propose sophisticated vector composition methods ("tensor algebra") where the resulting vector keeps track of the order of words in the original phrase
- Very interesting developments, but for the moment:
  - it is not clear that they could ever possibly capture the richness of hierarchical relations

A different approach to compositionality with DSMs

- When sentences are built, word vectors are checked and updated to enforce various composition constraints, but they are not fused
- Given general semantic composition rules that combine words having very few types (e, t, etc.), DSMs can be used to check the "commonsense" plausibility of the combination
  - the Montagovian composition component might tell us that both *Tom eats the mouse* and *Tom eats sympathy* are false
  - DSMs will tell us that the latter is also highly unlikely
    - cf. use of DSMs to model selectional preferences

Outline

Co-composition

Pustejovsky (1995), and many others

- When words are composed, they tend to affect each other's meanings
  - *The horse runs* vs. *The water runs*
  - "The horse horse-like runs"
  - cf. an instance of context-sensitive interpretation of lexical items
- Erk & Padó (2008)
  - *run* vector in the context of *horse* is a (multiplicative or additive) combination of the *run* vector and a prototype vector that represents the typical verbs *horse* is a subject of
    - *run-in-the-context-of-horse* = $\overrightarrow{run} \cdot (\overrightarrow{gallop} + \overrightarrow{trot} + \dots)$
  - *horse* vector in the context of *run* is a (multiplicative or additive) combination of the *horse* vector and a prototype vector that represents the typical subjects of *run*
    - *horse-in-the-context-of-run* = $\overrightarrow{horse} \cdot (\overrightarrow{car} + \overrightarrow{lion} + \dots)$
  - similar to Kintsch (2001), but now the predicate and the argument vectors are not fused together

Erk & Padó (2008)

- Vectors computed on 100M word BNC corpus, MINIPAR-based dependency-links, MI-weighting
- Measure cosine similarity of vector that, according to various models, represent verb-in-context, to landmark verb vector (e.g., "slump-in-the-context-of-value" vector to *decline* vector)
- Various configurations for vector combination
  - **verb**: use verb out-of-context vector
  - **prototype**: use prototype vector built from vectors of verbs that typically occur with noun as subject
  - **combined**: multiply verb vector and noun-as-subject prototype verb vector
  - **power-combined**: same, but values of dimensions of noun-as-subject prototype verb vector are raised to a power of 20
  - **ML**: Mitchell and Lapata's method: multiply noun and verb vector

Predicting verb-in-context similarity judgments

- Mitchell & Lapata's (2007, ML) data-set
- 49 subjects produced similarity ratings on 1-7 scale for intransitive subject-verb sentence pairs, one with context-affected verb, one with "landmark" reference verb:

subject	*verb*	*landmark*	*judgment*
shoulder	slump	slouch	7
shoulder	slump	decline	2
value	slump	slouch	3
value	slump	decline	7

- Average inter-subject Spearman correlation (ρ): 40%!

Results

Correlation with human judgments

power-combined	27%
ML	24%
prototype	16%
combined	13%
verb	8%

- Similar results for a subset of the SEMEVAL07 lexical substitution task, where *power-combined* and *prototype* outperform *ML*

Outline

Two views of vector composition in DSMs

- Syntagmatic composition
  - given the complex expression *ab*, we compose $\overrightarrow{a}$ and $\overrightarrow{b}$ to form a vector $\overrightarrow{ab}$
    - the composed vector is the interpretation of ab
- Paradigmatic composition
  - given the complex expression *ab*, we compose $\overrightarrow{a}$ with the vectors $\overrightarrow{c_1} \ldots \overrightarrow{c_n}$ of expressions that are paradigmatically similar to *b*
    - the composed vector is the interpretation of a-in-the-context-of-b
  - promising to capture various effects of context-sentitivity (cf. coercion, subsectivity effects in adjectives, etc.)

Two views of vector composition in DSMs

- In formal semantics we have a clear idea of the interpretation of complex expressions, i.e. sentences
  - sentences denote truth-values ($< t >$) or propositions ($< w, t >$)
  - the denotation of the component expressions is their contributions to the computation of the sentence denotation (cf. Frege's context principle)
- In DSMs we have a clear idea of the interpretation of words
  - words are interpreted on distributional vectors
- We don't have clear intuitions of what a composed vector stands for semantically
  - $\overrightarrow{w}$ is the distributional contextual representation of *w*, but what does the composed vector of a sentence represent?

Compositionality and contextual effects

- Compositionality could be a very simple process, but it is complicated by the behaviour of lexical items in context (producing type mismatches)
  - non-intersectivity
    - *skillful politician* vs. *fast typist* vs. *stone lion*
  - coercion
    - *enjoy a book* vs. *begin a book*

Kamp & Partee (1995: 163)

"It would seem that part of knowing the meaning of a word should have to involve knowing how the basic meaning(s) could be stretched, shrunk, or otherwise revised in various ways when necessary; since the possible revisions are probably not finitely specifiable, such a conception of meaning would take us well beyond the normal conception of the lexicon as a finite list of finite specifications of idiosyncratic information about a particular lexsical items"

Compositional semantics and DSMs

- A possible division of labour between formal semantics and DSMs
  - formal semantics contributes with a solid model of meaning composition
    - specifies how semantic types should compose
  - DSMs integrates it with a model for context-sentive modulation of semantic types
    - solves semantic type-mismatches
    - cf. the notion of concept recalibration in Kamp & Partee (1985), the notion of contextual concept in Bosch (1995), etc.
- This would be like taking *the best of two worlds*
- DSMs could simplify the mechanisms (e.g. type-shiftings, coercion, etc.) that are usually required to solve context-induced type-mismatches
  - a challenge for DSMs: address really hard cases of context-effects for formal semantics

That's all, folks!