Semantic Composition with Quotient Algebras

Daoud Clarke[1] Rudi Lutz[2] David Weir[2]

[1]Department of Computer Science
University of Hertfordshire

[2]Department of Informatics
University of Sussex

Geometric Models of Natural Language Semantics, 2010

Overview

Distributional Semantics

- Hypothesis of Harris (1968)
- LSA, distributional similarity etc.
- Many applications

Example

Corpus

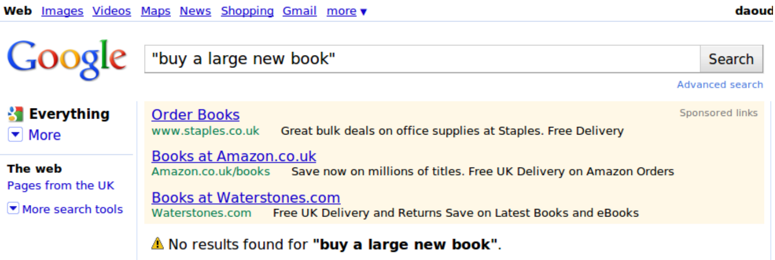
see big city
modernise city
see modern city
see red apple
buy apple
visit big apple
read big book
throw old small red book
buy large new book

	see	*buy*	*modernise*	*city*
modern	0	0	0	1
city	2	0	1	0

$$\widehat{modern} = (0, 0, 0, 1)$$
$$\widehat{city} = (2, 0, 1, 0)$$

Raw frequencies used for examples, but vectors could be obtained in any manner.

Data sparseness



Distributional semantic composition

- Strings of words represented by vectors
- Compose representations of words
- Related work considered:
  - Vector addition
  - Direct sum ($\oplus$)
  - Tensor product ($\otimes$)
  - More complex proposals (e.g. Clark et al.)

Comparing Composition Methods

- Addition is commutative
- $\oplus$ and $\otimes$ create vectors of different dimensionality
  - How to compare $u \otimes v$ with u?
- Composition defined by corpus data?

Direct Sum Definition

Direct sum of U and V defined as

$$U \oplus V = U \times V$$

Operations

$$\begin{aligned} (u_1, v_1) + (u_2, v_2) &= (u_1 + u_2, v_1 + v_2) \\ \alpha(u_1, v_1) &= (\alpha u_1, \alpha v_1) \end{aligned}$$

for $u_1, u_2 \in U$, $v_1, v2 \in V$ and $\alpha \in \mathbb{R}$.

Direct Sum Properties

- U and V embed into $U \oplus V$ by

$$\begin{aligned} u &\mapsto (u, 0) \\ v &\mapsto (0, v) \end{aligned}$$

- Write $u + v$ for $(u, 0) + (0, v)$
- Dimensionality

$$d(U \oplus V) = d(U) + d(V)$$

- $U \oplus V \cong V \oplus U$
    - Often treat $\oplus$ as commutative

Tensor Product

If U has orthonormal basis

$$\{u_1, u_2, \ldots u_n\}$$

and V has orthonormal basis

$$\{v_1, v_2, \ldots v_m\}$$

Tensor product basis

Then $U \otimes V$ has orthonormal basis

$$u_i \otimes v_j$$

for $1 \leq i \leq n$ and $1 \leq j \leq m$.

Example: Vector Addition and Direct Sum

$$
\begin{aligned}
\widehat{modern} &= (0, 0, 0, 1) \\
\widehat{city} &= (2, 0, 1, 0) \\
\widehat{modern} + \widehat{city} &= (2, 0, 1, 1) \\
\widehat{modern} \oplus \widehat{city} &\cong (0, 0, 0, 1, 2, 0, 1, 0)
\end{aligned}
$$

Example: Tensor Product

$$\widehat{modern} \otimes \widehat{city} \cong \begin{pmatrix} 0 \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix} \\ 0 \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix} \\ 0 \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix} \\ 1 \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Dimensionality Mismatch

- Tensor product is attractive
  - Smolensky (1990), Widdows (2008)
- Dimensionality of $u \otimes v$ is different from that of u
- How to define inner product so that

$$\langle \widehat{modern} \otimes \widehat{city}, \widehat{city} \rangle \neq 0?$$

Context-theoretic Semantics

- Abstract framework (Clarke, 2007)
- Generalise $\oplus$ and $\otimes$
- Vector space with bilinear product

$$
\begin{aligned}
x(y+z) &= xy + xz \\
(x+y)z &= xz + yz \\
(xy)z &= x(yz)
\end{aligned}
$$

- Associative algebra over a field

A General Method for Constructing Algebras

- Construct a free algebra (tensor algebra)

$$T(V) = \mathbb{R} \oplus V \oplus (V \otimes V) \oplus (V \otimes V \otimes V) \oplus \cdots$$

A General Method for Constructing Algebras

- Construct a free algebra (tensor algebra)

$$T(V) = \mathbb{R} \oplus V \oplus (V \otimes V) \oplus (V \otimes V \otimes V) \oplus \cdots$$

- Choose relations $u_1 = v_1, u_2 = v_2, \ldots$ we wish to hold

$$\Lambda = \{u_1 - v_1, u_2 - v_2, \ldots\}$$

A General Method for Constructing Algebras

- Construct a free algebra (tensor algebra)

$$T(V) = \mathbb{R} \oplus V \oplus (V \otimes V) \oplus (V \otimes V \otimes V) \oplus \cdots$$

- Choose relations $u_1 = v_1, u_2 = v_2, \ldots$ we wish to hold

$$\Lambda = \{u_1 - v_1, u_2 - v_2, \ldots\}$$

- Construct ideal I generated by Λ

A General Method for Constructing Algebras

- Construct a free algebra (tensor algebra)

$$T(V) = \mathbb{R} \oplus V \oplus (V \otimes V) \oplus (V \otimes V \otimes V) \oplus \cdots$$

- Choose relations $u_1 = v_1, u_2 = v_2, \ldots$ we wish to hold

$$\Lambda = \{u_1 - v_1, u_2 - v_2, \ldots\}$$

- Construct ideal I generated by Λ
- Take equivalence classes to get quotient algebra

$$T(V)/I$$

Quotient Algebra: Why?

- Vectors that were orthogonal in $T(V)$ can be non-orthogonal in $T(V)/I$
- Strings of different lengths can be compared

Quotient Algebra: How?

- An ideal I of an algebra A is a sub-vector space of A such that $xa \in I$ and $ax \in I$ for all $a \in A$ and all $x \in I$
- Congruence: $x \equiv y$ if $x - y \in I$
- Quotient algebra A/I formed from equivalence classes
- How to define the inner product $\langle [x], [y] \rangle$?

Approximating the Ideal

- Given basis B, define Λ_k ($k = 0, 1, 2, \ldots$) inductively:

$$\begin{aligned}\Lambda_0 &= \Lambda \\ \Lambda_k &= \Lambda_{k-1} \cup \{(e_i \otimes \Lambda_{k-1}) | e_i \in B\} \\ &\quad \cup \{(\Lambda_{k-1} \otimes e_i) | e_i \in B\}\end{aligned}$$

- G_k = vector space generated by Λ_k

Computing the Inner Product

- $T(V)/G_k$ is isomorphic to $G_k^\perp$
- Project x onto orthogonal complement $G_k^\perp$
  - Canonical representation x'_k of equivalence class $[x]_k$
  - Use Gram-Schmidt to compute projection
- Define $\langle [x]_k, [y]_k \rangle_k = \langle x'_k, y'_k \rangle$

Data-driven Composition

- Use a treebank
- For each rule $\pi : X \rightarrow X_1 \ldots X_r$ with head X_h we add vectors

$$\lambda_{\pi,i} = e_i - \widehat{X}_1 \otimes \ldots \otimes \widehat{X}_{h-1} \otimes e_i \otimes \widehat{X}_{h+1} \otimes \ldots \otimes \widehat{X}_r$$

  for each basis element e_i of V_{X_h} to the generating set.
  - $\widehat{X}$ is the sum over all individual vectors of subtrees rooted with X
- Preserve meaning of head

Example

Example Corpus

see big city
modernise city
see modern city
see red apple
buy apple
visit big apple
read big book
throw old small red book
buy large new book

Grammar:

$$\begin{aligned} \mathrm{N}' &\rightarrow \mathrm{Adj}\ \mathrm{N}' \\ \mathrm{N}' &\rightarrow \mathrm{N} \end{aligned}$$

Generating set Λ:

$$\lambda_i = e_i - \widehat{\mathrm{Adj}} \otimes e_i$$

$$\widehat{\mathrm{Adj}} = 2e_{apple} + 6e_{book} + 2e_{city}$$

where e_i ranges over basis vectors for noun contexts.

Example: Computing G_k

- Define $ST = \{st : s \in S \text{ and } t \in T\}$
- B is orthonormal basis for underlying vector space
- $G_0 = \Lambda$ is set of all $\lambda_i = e_i - \widehat{\mathrm{Adj}} \otimes e_i$
- $G_1 = (B \cup \{1\})G_0(B \cup \{1\})$
- $G_2 = (BB \cup B \cup \{1\})G_0(BB \cup B \cup \{1\})$

Example: Cosine Similarities

	apple	*big apple*	*red apple*	*city*	*big city*	*red city*	*book*	*big book*	*red book*
apple	1.0	0.26	0.24	0.52	0.13	0.12	0.33	0.086	0.080
big apple		1.0	0.33	0.13	0.52	0.17	0.086	0.33	0.11
red apple			1.0	0.12	0.17	0.52	0.080	0.11	0.33
city				1.0	0.26	0.24	0.0	0.0	0.0
big city					1.0	0.33	0.0	0.0	0.0
red city						1.0	0.0	0.0	0.0
book							1.0	0.26	0.24
big book								1.0	0.33
red book									1.0

Example: Cosine Similarities

	apple	*big apple*	*red apple*	*city*	*big city*	*red city*	*book*	*big book*	*red book*
apple	1.0	0.26	0.24	0.52	0.13	0.12	0.33	0.086	0.080
big apple		1.0	0.33	0.13	0.52	0.17	0.086	0.33	0.11
red apple			1.0	0.12	0.17	0.52	0.080	0.11	0.33
city				1.0	0.26	0.24	0.0	0.0	0.0
big city					1.0	0.33	0.0	0.0	0.0
red city						1.0	0.0	0.0	0.0
book							1.0	0.26	0.24
big book								1.0	0.33
red book									1.0

Example: Cosine Similarities

	apple	*big apple*	*red apple*	*city*	*big city*	*red city*	*book*	*big book*	*red book*
apple	1.0	0.26	0.24	0.52	0.13	0.12	0.33	0.086	0.080
big apple		1.0	0.33	0.13	0.52	0.17	0.086	0.33	0.11
red apple			1.0	0.12	0.17	0.52	0.080	0.11	0.33
city				1.0	0.26	0.24	0.0	0.0	0.0
big city					1.0	0.33	0.0	0.0	0.0
red city						1.0	0.0	0.0	0.0
book							1.0	0.26	0.24
big book								1.0	0.33
red book									1.0

Example: Cosine Similarities

	apple	*big apple*	*red apple*	*city*	*big city*	*red city*	*book*	*big book*	*red book*
apple	1.0	0.26	0.24	0.52	0.13	0.12	0.33	0.086	0.080
big apple		1.0	0.33	0.13	0.52	0.17	0.086	0.33	0.11
red apple			1.0	0.12	0.17	0.52	0.080	0.11	0.33
city				1.0	0.26	0.24	0.0	0.0	0.0
big city					1.0	0.33	0.0	0.0	0.0
red city						1.0	0.0	0.0	0.0
book							1.0	0.26	0.24
big book								1.0	0.33
red book									1.0

Future Work

- Use a big corpus
  - Restrict to finite dimensionality
- Other ways of choosing Λ
  - Lexicalised formalisms
  - Incorporate non-compositionality

Summary

- Tensor product composition
  - No way to compare strings of different lengths

Summary

- Tensor product composition
  - No way to compare strings of different lengths
- Quotient algebras
  - Introduce relations
  - Make strings of different lengths comparable
  - Relations can come from treebank data

Acknowledgments and References

Thanks to Metrica for supporting the research and Peter Hines, Stephen Clark, Peter Lane and Paul Hender for useful discussions.

- Daoud Clarke
  Context-theoretic Semantics for Natural Language
  DPhil thesis, University of Sussex
- Stephen Clark, Bob Coecke and Mehrnoosh Sadrzadeh
  A Compositional Distributional Model of Meaning
  QI-2008 2000.