

# Compositionality Detection using a Corpus Driven Approach: How to distinguish "couch potato" from "roast potato"

Siva Reddy<sup>1,2</sup>, Suresh Manandhar<sup>1</sup>, Diana McCarthy<sup>2</sup>

<sup>1</sup>Artificial Intelligence Group,  
Department of Computer Science,  
University of York, UK

<sup>2</sup>Lexical Computing Ltd., UK

2<sup>nd</sup> International Sketch Engine Workshop  
March 16 2011

# Outline

- 1 Compositionality
- 2 Goal of this work
- 3 Background
- 4 Our Approach
- 5 Evaluation

# Multi-word

- A sequence of two or more words describing a meaning together.
- Compound Nouns
  - credit card
  - leather jacket
- Phrasal Verbs
  - look up
  - get over
- Idiomatic expressions
  - kick the bucket
  - spill the beans

# Compositionality

Given meanings of

- couch
- roast
- potato

# Compositionality

Given meanings of

- couch
- roast
- potato

Can we interpret the meanings of

- couch potato
- roast potato

# Couch Potato



# Roast Potato



# Compositionality

## Compositional Multi-words

- $m(\text{"A B"}) = m(\text{A}) \oplus m(\text{B})$
- e.g. water tank, post man, roast potato
- caution: subjective task
- cracking " $\oplus$ " is a miracle

# Compositionality

## Compositional Multi-words

- $m(\text{"A B"}) = m(\text{A}) \oplus m(\text{B})$
- e.g. water tank, post man, roast potato
- caution: subjective task
- cracking " $\oplus$ " is a miracle

## Non-Compositional multi-words

- think tank, smoking gun, apple polisher, couch potato

# Importance of compositionality detection

## Dictionary Building: Lexicography and Terminology

- A good dictionary
  - includes non-compositional multi-words
  - does not have redundant information

## Machine Translation

- goose egg  $\neq$  Gänseei
- goose egg  $\rightarrow$  unwichtig
- Compositionality detection in a given context. Much harder.

## Word Tokenization

- Search engines

# Goal of this work

*Goal:* Identify compositional and non-compositional multi-words for a given language

- My focus is on *compound nouns*
- A sequence of nouns is treated as a *multi-word*
- Vast research on identifying multi-words but not on compositionality detection

# Goal of this work

*Goal:* Identify compositional and non-compositional multi-words for a given language

- My focus is on *compound nouns*
- A sequence of nouns is treated as a *multi-word*
- Vast research on identifying multi-words but not on compositionality detection

## Unsupervised corpus-based approach

- Huge corpora for many languages available in Sketch Engine
- Sketch Engine provides additional resources like word sketches, distributional thesaurus

# Background: Semantics from corpus

Distributional Hypothesis (Harris, 1954)

Words that occur in similar contexts tend to have similar meanings

# Background: Semantics from corpus

## Distributional Hypothesis (Harris, 1954)

Words that occur in similar contexts tend to have similar meanings

- e.g. Tree and Plant, Tea and Coffee, Bus and Vehicle

## Background: Semantics from corpus

### Distributional Hypothesis (Harris, 1954)

Words that occur in similar contexts tend to have similar meanings

- e.g. Tree and Plant, Tea and Coffee, Bus and Vehicle

### Other variations: (Firth, 1957)

You shall know a word by the company it keeps

## Background: Semantics from corpus

### Distributional Hypothesis (Harris, 1954)

Words that occur in similar contexts tend to have similar meanings

- e.g. Tree and Plant, Tea and Coffee, Bus and Vehicle

### Other variations: (Firth, 1957)

You shall know a word by the company it keeps

### Bag of words hypothesis

Two documents tend to be similar if they have similar distribution of similar words

# Vector Space Models (VSMs) of Semantics

- **Interpret semantics using VSM**
  - Backbone: Distributional Hypothesis

# Vector Space Models (VSMs) of Semantics

- **Interpret semantics using VSM**
  - Backbone: Distributional Hypothesis
- Text entity (we are interested in) as a Vector (point) in dimensional space.
- Context of the entity as dimensions

# Vector Space Models (VSMs) of Semantics

- **Interpret semantics using VSM**

- Backbone: Distributional Hypothesis
- Text entity (we are interested in) as a Vector (point) in dimensional space.
- Context of the entity as dimensions
- Existing methods represent knowledge in VSMs mainly in three types (Turney and Pantel, 2010)
  - term-document
  - term-context
  - pair-pattern

# Term-Document: (Salton et al., 1975)

Create a word-by-document matrix

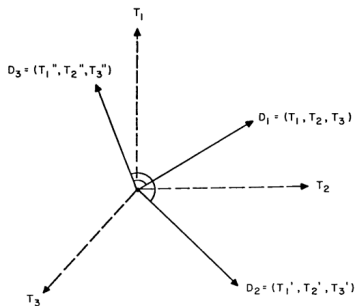
	d1	d2	d3	d4	d5	d6	d7	d8	d9
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	0	0	0	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

d1: **Human** machine **interface** for Lab ABC **computer** applications

---

<sup>1</sup>Image courtesy: (Landauer et al., 1998)

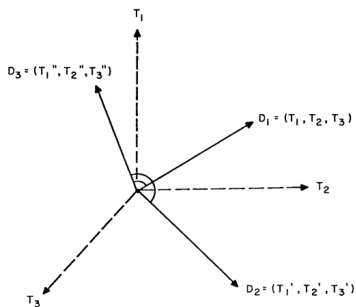
# Term-Document: (Salton et al., 1975)



2

<sup>2</sup>Image courtesy: (Salton et al., 1975)

# Term-Document: (Salton et al., 1975)



2

Document similarity can be found using Cosine similarity

- $$\text{sim}(D1, D2) = \frac{D1 \cdot D2}{\|D1\| \|D2\|}$$

<sup>2</sup>Image courtesy: (Salton et al., 1975)

## Term-Context: Word Space Model

	<b>buy</b>	<b>rent</b>	<b>sell</b>	<b>book</b>
house	50	60	38	0
apartment	60	100	36	0
room	0	40	0	100
suite	0	20	0	80

Words are represented as a vector build from context words

- I *rent* a *house*.
- I *bought* an *apartment*.
- I *booked* a *room*.

# Semantics of larger entities

*How to interpret semantics of larger entities?*

# Semantics of larger entities

*How to interpret semantics of larger entities?*

The distributional way

	turn	photon	sign	noise	speed
<b>Traffic</b>	5	0	3	10	15
<b>Light</b>	2	15	3	4	20
<i>TrafficLight</i> <sub>Dist</sub>	10	0	15	3	10

# Semantics of larger entities

*How to interpret semantics of larger entities?*

The distributional way

	turn	photon	sign	noise	speed
<b>Traffic</b>	5	0	3	10	15
<b>Light</b>	2	15	3	4	20
<i>TrafficLight</i> <sub>Dist</sub>	10	0	15	3	10

*How to interpret semantics of larger entities from its constituents?*

## Semantics of larger entities

*How to interpret semantics of larger entities?*

The distributional way

	turn	photon	sign	noise	speed
<b>Traffic</b>	5	0	3	10	15
<b>Light</b>	2	15	3	4	20
<i>TrafficLight</i> <sub>Dist</sub>	10	0	15	3	10

*How to interpret semantics of larger entities from its constituents?*

The Principle of Compositionality:(Partee et al., 1990)

The meaning of a compound expression is a function of, and only of, the meaning of its parts and the way in which the parts are combined.

## Idea exploited by many:

### Two ways of computing a multi-word's meaning

- Distributional way: The true meaning vector of the multi-word
- Compositionality function: A way of estimating the meaning vector of the multi-word from the meaning vectors of its parts

## Idea exploited by many:

### Two ways of computing a multi-word's meaning

- Distributional way: The true meaning vector of the multi-word
- Compositionality function: A way of estimating the meaning vector of the multi-word from the meaning vectors of its parts

### How similar are these two vectors?

- If they are very close, i.e. meaning of the multi-word can be computed from the meaning of the parts, the multi-word is compositional
- else non-compositional

## Idea exploited by many:

### Two ways of computing a multi-word's meaning

- Distributional way: The true meaning vector of the multi-word
- Compositionality function: A way of estimating the meaning vector of the multi-word from the meaning vectors of its parts

### How similar are these two vectors?

- If they are very close, i.e. meaning of the multi-word can be computed from the meaning of the parts, the multi-word is compositional
- else non-compositional

*Bingo!!*

# Category 1: Katz and Giesbrecht (2006); Giesbrecht (2009)

- Build
  - $CouchPotato_{Dist}$
  - $CouchPotato_{Comp}$  i.e. **Couch**  $\oplus$  **Potato**
- $sim(CouchPotato_{Dist}, CouchPotato_{Comp})$ 
  - if  $sim > thrsh$ : multi-word is compositional
  - else: multi-word is non-compositional

# Category 1: Katz and Giesbrecht (2006); Giesbrecht (2009)

- Build
  - $CouchPotato_{Dist}$
  - $CouchPotato_{Comp}$  i.e. **Couch**  $\oplus$  **Potato**
- $sim(CouchPotato_{Dist}, CouchPotato_{Comp})$ 
  - if  $sim > thrsh$ : multi-word is compositional
  - else: multi-word is non-compositional
- Pitfalls observed:
  - Threshold highly varies
  - 48 % accuracy

## Category 2: (Baldwin et al., 2003; Bannard et al., 2003)

Build distributional vectors of

- $CouchPotato_{Dist}$
- $Couch$
- $Potato$

$sim(CouchPotato_{Dist}, Potato)$

- if  $sim > thrsh$ : multi-word is compositional
- else: multi-word is non-compositional

## Category 2: (Baldwin et al., 2003; Bannard et al., 2003)

Build distributional vectors of

- $CouchPotato_{Dist}$
- $Couch$
- $Potato$

$sim(CouchPotato_{Dist}, Potato)$

- if  $sim > thrsh$ : multi-word is compositional
- else: multi-word is non-compositional

Pitfalls Observed:

- Was able to capture *type-of* relations only
- *Threshold* highly varies
- Moderate results: 51 % accuracy

# Observations

## Observations

- Threshold highly varies. Reported by everyone
- What might be the possible reasons?

# Observations

## Observations

- Threshold highly varies. Reported by everyone
- What might be the possible reasons?

## Polysemy

- Skewed nature of senses
- $RiverBank_{dist}$  is not similar to  $Bank_{dist}$
- Either of the above approaches fail
- Focus of this presentation is to overcome this using Sketch Engine

# Observations

## Observations

- Threshold highly varies. Reported by everyone
- What might be the possible reasons?

## Polysemy

- Skewed nature of senses
- $RiverBank_{dist}$  is not similar to  $Bank_{dist}$
- Either of the above approaches fail
- Focus of this presentation is to overcome this using Sketch Engine

## Continuum (McCarthy et al., 2003)

- There is no hard boundary to say if a multi-word is compositional

# Concordance of Sketch Engine as Term-Context Matrix

the idea of a much worse prison ; where No **light** , but rather darkness visible , Served  
 tabernacle-work over the stalls carved in a **light** and elegant manner . St. John 's , which  
 There was a general , unrelieved , dull **light** ; so that , unless when looking at your  
 He half opened one of them , and as the **light** poured in , looked round with mournful  
 beauty heightened by the aid of brilliant **lights** , of costly jewels , and all the pride  
 use the cycle paths and have good bright **lights** , then you should have no problems . Bus  
 . I think it puts business in a very bad **light** . Alan Sugar does everyone a great disservice  
 framework , Tati became an influential guiding **light** for the generations of comedians and filmmakers  
 morning - it 's night It 's dark - it 's **light** It 's raining - it 's sunny life 's serious  
 or feeling low M Baird 167 The Northern **lights** and Mackie 's means home sweet home to  
 This investigation is intended to bring to **light** some reasons for connecting the notion  
 24 hours a day and the proprietors keep **light** security , particularly a local rent a cop  
 form with the only pleasantness being the **light** white fluffy foam of the recently sumped  
 I thought his material had all seen the **light** of day . TK 's mentor , Henry Stone sent  
 1973 . I am amazed this has n't seen the **light** of day . It is wonderful , and definitely  
 legal issues , that it would never see the **light** of day . Frank has taken the reigns , as  
 quite rightly so . Bright and breezy and **lit** up a few dancefloors as well as receiving  
 requested . â Ć Ideal as a clip-on book **light** â Ć Reaches places other torches can  
 cover at the end of the arm . The Flexi **Light** requires two AAA batteries ( not included  
 and clips in for compact storage . Flexi **Light** FAQ 's : Q ) Hi , what bulb should I use

Much powerful than conventional Term-Context matrix

## Ignored fact: Words are polysemous

### Current trend: Prototype Vectors

Currently most methods represent each word as a single vector i.e. a prototype vector for each word.

## Ignored fact: Words are polysemous

### Current trend: Prototype Vectors

Currently most methods represent each word as a single vector i.e. a prototype vector for each word.

### Light occur in many contexts

- Quantum theory, Optics, Bulbs and Traffic
- Not all contexts are relevant for building compositional vectors.
- Light is noisy  $\implies$  *TrafficLight*<sub>Comp</sub> is noisy

## Ignored fact: Words are polysemous

### Current trend: Prototype Vectors

Currently most methods represent each word as a single vector i.e. a prototype vector for each word.

### Light occur in many contexts

- Quantum theory, Optics, Bulbs and Traffic
- Not all contexts are relevant for building compositional vectors.
- Light is noisy  $\implies$  *TrafficLight*<sub>Comp</sub> is noisy

### Exemplars of Light

'interest-n': 1.0, 'round-n': 1.0, 'open-v': 1.0

'business-n': 1.0, 'bad-j': 1.0, 'put-v': 1.0

'framework-n': 1.0, 'generation-n': 1.0, 'technique-n': 1.0, 'follow-v': 1.0

'material-n': 1.0, 'day-n': 1.0, 'complete-j': 1.0

# Proposed Solution

*Prototype vectors are more noisy*  
*A need for refined vectors*

# Proposed Solution

*Prototype vectors are more noisy*  
*A need for refined vectors*

## Exemplar-based Vector Space Model

- Select (examples) exemplars of *Light* which have similar context of *Traffic*
- Prunes out irrelevant exemplars
- Use selected exemplars to build  $Light_{Traffic}$
- Motivated from the work of Erk and Pado (2010)

# Proposed Solution

*Prototype vectors are more noisy  
A need for refined vectors*

## Exemplar-based Vector Space Model

- Select (examples) exemplars of *Light* which have similar context of *Traffic*
- Prunes out irrelevant exemplars
- Use selected exemplars to build  $Light_{Traffic}$
- Motivated from the work of Erk and Pado (2010)

*How to select (examples) exemplars of Light which have similar context of Traffic??*

# First order co-occurrences of *traffic* from Sketch Engine

<b>object_of</b> 18950 1.4	<b>and/or</b> 10005 0.8	<b>pp_along-i</b> 95 5.2	<b>n_modifier</b> 23201 2.5
divert 198 7.78	pedestrian 157 7.56	road 25 0.92	freight 512 8.63
slow 212 7.58	congestion 134 6.92	route 14 0.83	road 5520 8.63
block 319 7.5	pollution 168 6.42	street 9 0.62	air 2248 8.09
generate 711 7.34	noise 188 5.8		passenger 708 7.54
speed 149 7.25	parking 173 5.57	<b>pp_onto-i</b> 64 4.6	rush 211 7.45
motorise 93 7.17	freight 39 5.45	road 25 0.92	commuter 134 7.09
encrypt 83 6.79	roadwork 14 5.26	route 9 0.2	motor 360 6.94
rout 78 6.76	traffic 190 5.17		rail 240 6.27
direct 218 6.65	transportation 32 5.09	<b>pp_off-i</b> 54 4.4	network 874 6.0
calm 75 6.5	highway 38 5.05	road 31 1.23	motorway 85 5.96
congest 55 6.43	passenger 99 4.87		good 350 5.85
wheel 55 6.28	fume 14 4.73	<b>pp_through-i</b> 305 2.8	lorry 73 5.85
redirect 52 6.24	pedestrianisation 8 4.63	firewall 10 4.35	Internet 492 5.83
monitor 260 6.21	lorry 20 4.53	village 46 2.74	coal 117 5.75
increase 985 6.18	commuter 13 4.49	port 18 2.7	multicast 37 5.63
drive 407 6.12	motorway 20 4.38	tunnel 8 2.69	tourist 106 5.48
stop 330 6.1	transport 130 4.33		barge 40 5.27
reduce 698 6.08	TCP 11 4.33		data 143 5.25
induce 83 6.01	road 240 4.15		container 72 5.23

*Word Sketch of traffic to select exemplars of light*

## Similar Words to *Traffic* from Sketch Engine

Lemma	Score	Freq
<a href="#">transport</a>	0.362	134717
<a href="#">road</a>	0.339	324641
<a href="#">train</a>	0.336	114514
<a href="#">vehicle</a>	0.331	160671
<a href="#">bus</a>	0.322	131884
<a href="#">route</a>	0.312	168121
<a href="#">network</a>	0.311	262162
<a href="#">trade</a>	0.308	165216
<a href="#">market</a>	0.307	379176
<a href="#">travel</a>	0.306	115459
<a href="#">communication</a>	0.3	171501
<a href="#">flow</a>	0.299	77846
<a href="#">station</a>	0.295	175788
<a href="#">operation</a>	0.295	198053
<a href="#">car</a>	0.293	419404
<a href="#">access</a>	0.289	376109
<a href="#">business</a>	0.287	700710
<a href="#">speed</a>	0.286	146544
<a href="#">sale</a>	0.285	239489

*Not only context words of Traffic but also similar words to Traffic are useful [courtesy: Diana]*

# Exemplar-based Composition

## Exemplars of *LightTraffic*

'speed-n': 4.0, 'create-v': 1.0, 'mass-n': 1.0

'road-n': 2.0, 'good-j': 1.0, 'white-j': 3.0

'street-n': 1.0, 'road-n': 2.0, 'limit-n': 1.0, 'sign-n': 1.0

'road-n': 2.0, 'side-n': 1.0, 'wrong-j': 1.0, 'drive-v': 1.0

'bright-j': 15.0, 'day-n': 15.0

# Exemplar-based Composition

## Exemplars of *LightTraffic*

'speed-n': 4.0, 'create-v': 1.0, 'mass-n': 1.0

'road-n': 2.0, 'good-j': 1.0, 'white-j': 3.0

'street-n': 1.0, 'road-n': 2.0, 'limit-n': 1.0, 'sign-n': 1.0

'road-n': 2.0, 'side-n': 1.0, 'wrong-j': 1.0, 'drive-v': 1.0

'bright-j': 15.0, 'day-n': 15.0

*Build vector of Light using the above exemplars: LightTraffic*

# Exemplar-based Composition

## Exemplars of $Light_{Traffic}$

'speed-n': 4.0, 'create-v': 1.0, 'mass-n': 1.0

'road-n': 2.0, 'good-j': 1.0, 'white-j': 3.0

'street-n': 1.0, 'road-n': 2.0, 'limit-n': 1.0, 'sign-n': 1.0

'road-n': 2.0, 'side-n': 1.0, 'wrong-j': 1.0, 'drive-v': 1.0

'bright-j': 15.0, 'day-n': 15.0

*Build vector of  $Light$  using the above exemplars:  $Light_{Traffic}$*

## Exemplar-based Composition

$TrafficLight_{Comp} = Traffic_{Light} \oplus Light_{Traffic}$

# Traffic Light: Evaluation

- ukWaC in Sketch Engine
- Traffic: 97492 examples
- Light: 316133 examples
- *TrafficLight<sub>Dist</sub>*: 6730 examples

## Traffic Light: Evaluation

- ukWaC in Sketch Engine
- Traffic: 97492 examples
- Light: 316133 examples
- $TrafficLight_{Dist}$ : 6730 examples

### Prototype-based Model

- $sim(TrafficLight_{Dist}, TrafficLight_{Comp}^{Prt}) = 0.434$

## Traffic Light: Evaluation

- ukWaC in Sketch Engine
- Traffic: 97492 examples
- Light: 316133 examples
- $TrafficLight_{Dist}$ : 6730 examples

### Prototype-based Model

- $sim(TrafficLight_{Dist}, TrafficLight_{Comp}^{Prt}) = 0.434$

### Exemplar-based Model

- $Traffic_{Light}$ : 1949 exemplars
- $Light_{Traffic}$ : 6322 exemplars
- Just 1% of the examples
- $sim(TrafficLight_{Dist}, TrafficLight_{Comp}^{Exm}) = 0.509$

# Couch Potato: Evaluation

- Couch: 5103 examples
- Potato: 23277 examples
- *CouchPotato<sub>Dist</sub>*: 407 examples

# Couch Potato: Evaluation

- Couch: 5103 examples
- Potato: 23277 examples
- $CouchPotato_{Dist}$ : 407 examples

## Prototype-based Model

- $sim(CouchPotato_{Dist}, CouchPotato_{Comp}^{Prt}) = 0.134$

# Couch Potato: Evaluation

- Couch: 5103 examples
- Potato: 23277 examples
- $CouchPotato_{Dist}$ : 407 examples

## Prototype-based Model

- $sim(CouchPotato_{Dist}, CouchPotato_{Comp}^{Prt}) = 0.134$

## Exemplar-based Model

- $Couch_{Potato}$ : 41 exemplars
- $Potato_{Couch}$ : 5 exemplars
- $sim(CouchPotato_{Dist}, CouchPotato_{Comp}^{Exm}) = 0.005$

# Observations and Evaluation

## Observations

Exemplar-based composition using Sketch Engine

- Rewards compositional multi-words
- Penalizes non-compositional multi-words
- This is what we want!!

## Evaluation [along with Diana]

- 200 Mechanical turkers annotated 90 words for multi-word compositionality
- To be evaluated on this data
- More findings about the Principle of Compositionality

## Other aspects

- Compositionality function and its parameters
- Parameter estimation
- Principle of Compositionality: Its pitfalls
- Anatomy aware model of compositionality detection

# Summary

- Compositionality and its importance
- Vector Space Models for Compositionality Detection
- Polysemy is a major problem
- Exemplar-based VSM using Sketch Engine

## Bibliography I

Baldwin, T., Bannard, C., Tanaka, T., and Widdows, D. (2003). An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, pages 89–96, Morristown, NJ, USA. Association for Computational Linguistics.

Bannard, C., Baldwin, T., and Lascarides, A. (2003). A statistical approach to the semantics of verb-particles. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, MWE '03, pages 65–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

Erk, K. and Pado, S. (2010). Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 92–97, Uppsala, Sweden. Association for Computational Linguistics.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. 1952-59:1–32

## Bibliography II

- Giesbrecht, E. (2009). In search of semantic compositionality in vector spaces. In *Proceedings of the 17th International Conference on Conceptual Structures: Conceptual Structures: Leveraging Semantic Technologies*, ICCS '09, pages 173–184, Berlin, Heidelberg. Springer-Verlag.
- Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.
- Katz, G. and Giesbrecht, E. (2006). Automatic identification of non-compositional multi-word expressions using latent semantic analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, MWE '06, pages 12–19, Morristown, NJ, USA. Association for Computational Linguistics.
- Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

## Bibliography III

- McCarthy, D., Keller, B., and Carroll, J. (2003). Detecting a continuum of compositionality in phrasal verbs. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, MWE '03, pages 73–80, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Partee, B. H., Meulen, T. A. G., and Wall, R. (1990). *Mathematical Methods in Linguistics (Studies in Linguistics and Philosophy)*. Springer.
- Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18:613–620.
- Turney, P. D. and Pantel, P. (2010). From frequency to meaning: vector space models of semantics. *J. Artif. Int. Res.*, 37:141–188.