Compositionality Detection using Vector Space Model:
How to distinguish "couch potato" from "roast potato"

Siva Reddy

Artificial Intelligence Group
Department of Computer Science
University of York

AIG Seminar
Feb 02 2011

THE UNIVERSITY of York

Outline

THE UNIVERSITY *of York*

Background: Foundations of Semantics

Distributional Hypothesis (Harris, 1954)

Words that occur in similar contexts tend to have similar meanings

Background: Foundations of Semantics

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: Foundations of Semantics

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: Foundations of Semantics

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

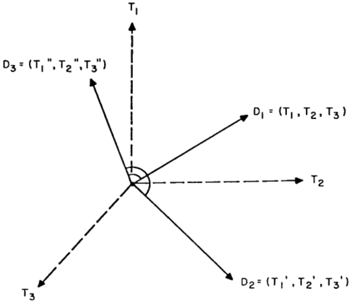
	d1	d2	d3	d4	d5	d6	d7	d8	d9
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	0	0	0	0
graph	0	0	0	0	0	0	1	1	1
minors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

[1]

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

[1] Image courtesy: (Landauer et al., 1998)

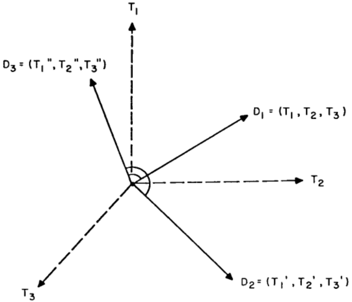
Term-Document: (Salton et al., 1975)



[2]

[2] Image courtesy: (Salton et al., 1975)

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



[2]

Document similarity can be found using Cosine similarity

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

[2] Image courtesy: (Salton et al., 1975)

Term-Context: Word Space Model

	buy	rent	sell	book
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*.

Compositionality

How to interpret semantics of larger entities?

Compositionality

How to interpret semantics of larger entities?

The distributional way

	turn	photon	sign	noise	speed
Traffic	5	0	3	10	15
Light	2	15	3	4	20
TrafficLight$_{\mathbf{Dist}}$	10	0	15	3	10

Compositionality

How to interpret semantics of larger entities?

The distributional way

	turn	photon	sign	noise	speed
Traffic	5	0	3	10	15
Light	2	15	3	4	20
TrafficLight$_{\mathbf{Dist}}$	10	0	15	3	10

How to interpret semantics of larger entities from its constituents?

Compositionality

How to interpret semantics of larger entities?

The distributional way

	turn	photon	sign	noise	speed
Traffic	5	0	3	10	15
Light	2	15	3	4	20
TrafficLight$_{\textbf{Dist}}$	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 complex expression is a function of the meaning of its parts and of the syntactic rules by which they are combined.

Outline

THE UNIVERSITY of York

Compositionality Functions

- Compositionality function $V \oplus W$
- Existing compositionality functions (Mitchell and Lapata, 2008; Widdows, 2008)
  - Addition
    - VW[i] : V[i] + W[i]
    - $dim(VW) = dim(V) = dim(W)$
    - Widely used and works in most information retrieval systems

THE UNIVERSITY of York

Compositionality Functions

- Compositionality function $V \oplus W$
- Existing compositionality functions (Mitchell and Lapata, 2008; Widdows, 2008)
  - Addition
    - VW[i] : V[i] + W[i]
    - $dim(VW) = dim(V) = dim(W)$
    - Widely used and works in most information retrieval systems
  - Multiplication
    - VW[i]= V[i] * W[i]
    - $dim(VW) = dim(V) = dim(W)$
    - Paraphrase detection

THE UNIVERSITY of York

Compositionality functions

- Complex compositionality functions: Tensor Product
  - VW(i,j) = V [i] * W [j] i.e. a Tensor (matrix of rank two)
  - Transforms into a new dimensional space
  - $dim(VW) = dim(V) \times dim(W)$
  - Can capture hidden relations between vectors
  - $Moscow : X :: London - Britain$

Compositionality functions

- Complex compositionality functions: Tensor Product
  - VW(i,j) = V [i] * W [j] i.e. a Tensor (matrix of rank two)
  - Transforms into a new dimensional space
  - $dim(VW) = dim(V) \times dim(W)$
  - Can capture hidden relations between vectors
  - $Moscow : X :: London - Britain$
- Vector addition is the most competitive among the above functions (Guevara:10)

THE UNIVERSITY *of York*

How good is our compositional vector?

	turn	photon	sign	noise	speed
Traffic	5	0	3	10	15
Light	2	15	3	4	20
$\mathit{TrafficLight}_{\mathbf{Dist}}$	10	0	15	3	10

- $\mathit{TrafficLight}_{Comp} = \mathbf{Traffic} \oplus \mathbf{Light}$

THE UNIVERSITY *of York*

How good is our compositional vector?

	turn	photon	sign	noise	speed
Traffic	5	0	3	10	15
Light	2	15	3	4	20
TrafficLight$_{\textbf{Dist}}$	10	0	15	3	10

- $TrafficLight_{Comp} = \textbf{Traffic} \oplus \textbf{Light}$
- One possible evaluation:
  - How similar is $TrafficLight_{Comp}$ to $TrafficLight_{\textbf{Dist}}$?
  - Higher the similarity better is our model.

My current focus: Additive Model

	turn	photon	sign	noise	speed
Traffic	5	0	3	10	15
Light	2	15	3	4	20
$\mathit{TrafficLight}_{\mathbf{Dist}}$	10	0	15	3	10

$$\mathit{TrafficLight}_{Comp} = A\ \textbf{Traffic} + B\ \textbf{Light}$$

so that

$$\mathit{sim}(\mathit{TrafficLight}_{Comp}, \mathit{TrafficLight}_{Dist}) = 1$$

How to determine weights A and B in the equation?

Outline

THE UNIVERSITY of York

Mitchell and Lapata (2008)

- A and B are scalars.
- Trail and Error method on a development data-set.
- A= 20 B= 80
- Most methods use this.

Mitchell and Lapata (2008)

- A and B are scalars.
- Trail and Error method on a development data-set.
- A= 20 B= 80
- Most methods use this.

Machine Learning for linear models

- A and B are matrices
- Posed as a Multivariate regression problem (Guevara, 2010)
- Linear Algebra approximation (Zanzotto et al., 2010) (York)

Problem1: Degree of Composition

Words have varying degree of compositionality

- $sim(Traffic, TrafficLight_{Dist}) = 0.624$
- $sim(Light, TrafficLight_{Dist}) = 0.356$
- **Traffic** contributes more towards the meaning of **TrafficLight**

THE UNIVERSITY of York

Problem1: Degree of Composition

Words have varying degree of compositionality

- $sim(Traffic, TrafficLight_{Dist}) = 0.624$
- $sim(Light, TrafficLight_{Dist}) = 0.356$
- **Traffic** contributes more towards the meaning of **TrafficLight**

- $sim(Student, StudentNurse_{Dist}) = 0.238$
- $sim(Nurse, StudentNurse_{Dist}) = 0.893$
- **Nurse** contributes more towards the meaning of **StudentNurse**

THE UNIVERSITY of York

Conclusion1

Static weights don't work
A need for dynamic weights

Problem2: Words are polysemous

Definition: Prototype Vector

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

Problem2: Words are polysemous

Definition: Prototype Vector

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 $\Longrightarrow$ $\textit{TrafficLight}_{\textit{Comp}}$ is noisy

Problem2: Words are polysemous

Definition: Prototype Vector

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 $\Longrightarrow$ $\mathit{TrafficLight}_{Comp}$ 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

Problem2 and its solution

Prototype vectors are more noisy

A need for refined vectors

Problem2 and its solution

Prototype vectors are more noisy

A need for refined vectors

Exemplar-based Vectors

- 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)

THE UNIVERSITY of York

Problem2 and its solution

Prototype vectors are more noisy

A need for refined vectors

Exemplar-based Vectors

- 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??

Outline

Context of *traffic*: first order co-occurences

object_of	18950	1.4
divert	198	7.78
slow	212	7.58
block	319	7.5
generate	711	7.34
speed	149	7.25
motorise	93	7.17
encrypt	83	6.79
rout	78	6.76
direct	218	6.65
calm	75	6.5
congest	55	6.43
wheel	55	6.28
redirect	52	6.24
monitor	260	6.21
increase	985	6.18
drive	407	6.12
stop	330	6.1
reduce	698	6.08
induce	83	6.01

and/or	10005	0.8
pedestrian	157	7.56
congestion	134	6.92
pollution	168	6.42
noise	188	5.8
parking	173	5.57
freight	39	5.45
roadwork	14	5.26
traffic	190	5.17
transportation	32	5.09
highway	38	5.05
passenger	99	4.87
fume	14	4.73
pedestrianisation	8	4.63
lorry	20	4.53
commuter	13	4.49
motorway	20	4.38
transport	130	4.33
TCP	11	4.33
road	240	4.15

pp_along-i	95	5.2
road	25	0.92
route	14	0.83
street	9	0.62

pp_onto-i	64	4.6
road	25	0.92
route	9	0.2

pp_off-i	54	4.4
road	31	1.23

pp_through-i	305	2.8
firewall	10	4.35
village	46	2.74
port	18	2.7
tunnel	8	2.69

n_modifier	23201	2.5
freight	512	8.63
road	5520	8.63
air	2248	8.09
passenger	708	7.54
rush	211	7.45
commuter	134	7.09
motor	360	6.94
rail	240	6.27
network	874	6.0
motorway	85	5.96
good	350	5.85
lorry	73	5.85
Internet	492	5.83
coal	117	5.75
multicast	37	5.63
tourist	106	5.48
barge	40	5.27
data	143	5.25
container	72	5.23

Sketch Engine (`www.sketchengine.co.uk`) is used to extract these co-occurences

THE UNIVERSITY *of York*

Similar Words to *Traffic*: second-order co-occurences

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

Not only context words of Traffic but also similar words to Traffic are useful

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

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

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

$TrafficLight_{Comp} = A\ Traffic_{Light} + B\ Light_{Traffic}$

THE UNIVERSITY *of York*

TrafficLight: Evaluation

- $sim(TrafLight_{Dist}, TrafLight_{Comp})$ as the evaluation metric.
- UKWaC Corpus: UK Web as Corpus
- Equal weights i.e. A=B

TrafficLight: Evaluation

- $sim(TrafLight_{Dist}, TrafLight_{Comp})$ as the evaluation metric.
- UKWaC Corpus: UK Web as Corpus
- Equal weights i.e. A=B

Prototype-based Model

- $TrafficLight_{Dist}$: 7153 examples
- Traffic: 101454 examples
- Light: 333226 examples
- $sim(TrafficLight_{Dist}, {TrafLight_{Comp}}^{Prt}) = 0.635$

TrafficLight: Evaluation

- $sim(TrafLight_{Dist}, TrafLight_{Comp})$ as the evaluation metric.
- UKWaC Corpus: UK Web as Corpus
- Equal weights i.e. A=B

Prototype-based Model

- $TrafficLight_{Dist}$: 7153 examples
- Traffic: 101454 examples
- Light: 333226 examples
- $sim(TrafficLight_{Dist}, {TrafLight_{Comp}}^{Prt}) = 0.635$

Exemplar-based Model

- $Traffic_{Light}$: 2029 exemplars
- $Light_{Traffic}$: 6664 exemplars
- $sim(TrafficLight_{Dist}, {TrafLight_{Comp}}^{Exm}) = 0.683$

Conclusion2:

Compositionality benefits from Exemplar-based Vectors than Prototype-based

Outline

Compositionality Detection

Goal

Detect if a multi-word is compositional or not.

To benefit from the above conclusions

- Conclusion1: Static Weights (A, B) do not work
- Conclusion2: Exemplar-based Vectors are beneficial compared to Prototype-based Vectors

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

Multi-word Compositionality

Given meanings of

- Couch
- Roast
- Potato

Multi-word Compositionality

Given meanings of

- Couch
- Roast
- Potato

Can we interpret the meanings of

- Couch Potato
- Roast Potato

Couch Potato

Roast Potato

Multi-word Compositionality

A multi-word *"A B"* is Compositional

- if $meaning(A\ B) = meaning(A) \oplus meaning(B)$
- e.g. Roast Potato
- e.g. Post Man

Multi-word Compositionality

A multi-word *"A B"* is Compositional

- if $meaning(A\ B) = meaning(A) \oplus meaning(B)$
- e.g. Roast Potato
- e.g. Post Man

Non-Compositional

- Couch Potato
- Think Tank
- Smoking Gun
- Apple Polisher

Problem Definition

Given: Large Web Corpus of a language

- English: 15 billion word corpora
- German: 1 billion word corpora
- Telugu: 10 million word corpora

THE UNIVERSITY of York

Problem Definition

Given: Large Web Corpus of a language

- English: 15 billion word corpora
- German: 1 billion word corpora
- Telugu: 10 million word corpora

Goal: Identify compositional and non-compositional multi-words.

- My focus is on *compound nouns*
- A sequence of nouns is treated as a *multi-word*

Importance

Dictionary Building

- A good dictionary does not miss non-compositional multi-words

Machine Translation

- Non-compositional words should be treated as a single word
- goose egg $\neq$ Gänseei
- goose egg $\rightarrow$ unwichtig

Word Tokenization

- Search engines

Outline

1. Background
2. Compositionality
   - Compositionality Functions
   - Problems in Compositionality
   - Exemplar-based Composition
3. Compositionality Detection
   - Compositonality Detection
   - Previous Approaches
   - Problems
   - Proposed Approach

THE UNIVERSITY of York

Previous Approaches: Lin (1999)

Roast Potato

- substituting thesaurus entries of *Roast*
- *Fried* Potato
- *Hot* Potato
- *Crisp* Potato

Couch Potato

- *Sofa* Potato
- *Chair* Potato

Previous Approaches: Lin (1999)

Roast Potato

- substituting thesaurus entries of *Roast*
- *Fried* Potato
- *Hot* Potato
- *Crisp* Potato

Couch Potato

- *Sofa* Potato
- *Chair* Potato

not all that glitters is gold

- Fails for *Water Tank*
- *Drink* Tank
- 15.7 % accuracy reported

Compositionality Detection using VSM (Baldwin et al., 2003)

Step1: Build distributional vectors of

- $CouchPotato_{Dist}$
- $Potato$

Step2: Measure $sim(CouchPotato_{Dist}, Potato)$

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

Compositionality Detection using VSM (Baldwin et al., 2003)

Step1: Build distributional vectors of

- $CouchPotato_{Dist}$
- *Potato*

Step2: Measure $sim(CouchPotato_{Dist}, Potato)$

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

Pitfalls

- Was able to capture *type-of* relations only
- *Threshold* highly varies
  - Skewed nature of senses
  - *RiverBank* is not similar to *Bank*
- Moderate results: 51 % accuracy

THE UNIVERSITY *of York*

Katz and Giesbrecht (2006); Giesbrecht (2009)

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

Katz and Giesbrecht (2006); Giesbrecht (2009)

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

Outline

THE UNIVERSITY of York

Problem3:

Threshold for identifying if a multi-word is compositional highly varies

Possible Reasons

- There is no hard cut-off which every multi-word obey
- Polysemous nature of words

Problem4:

Existing methods try to identify two classes

- Compositional
- Non Compositional

Problem4:

Existing methods try to identify two classes

- Compositional
- Non Compositional

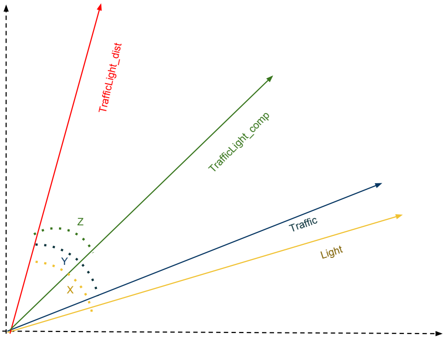
We found four classes: We manually created four data-sets

Meaning of multi-word “Word1 Word2” is related to

- Both Word1 and Word2
  - Roast Potato
- Only Word1 and not Word2
  - Couch Potato
- Only Word2 and not Word1
  - Zebra Crossing
- Neither Word1 nor Word2
  - Smoking Gun

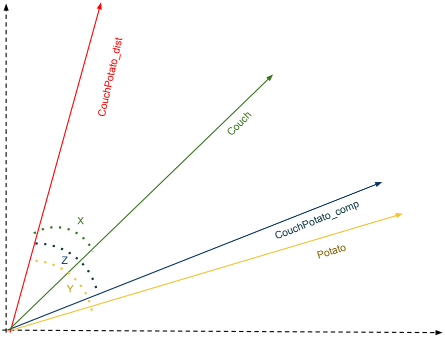
Outline

Solution to the above problems: Relative Threshold



- Relative Threshold: $sim(TrafLight_{Dist}, TrafLight_{Comp})$ relative to $sim(TrafLight_{Dist}, Traffic)$ and $sim(TrafLight_{Dist}, Light)$
- if Z < X and Z < Y, then multi-word is compositional i.e. Class1

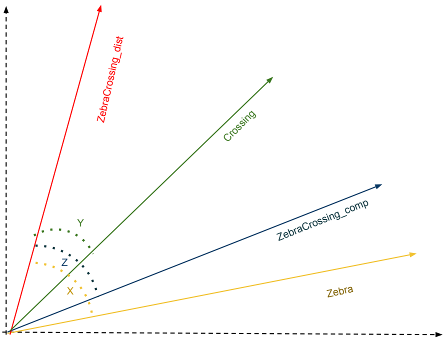
Semi-Compositional: Class2



- if Z > X and Z < Y, then multi-word is semi-compositional i.e. Class2

THE UNIVERSITY of York

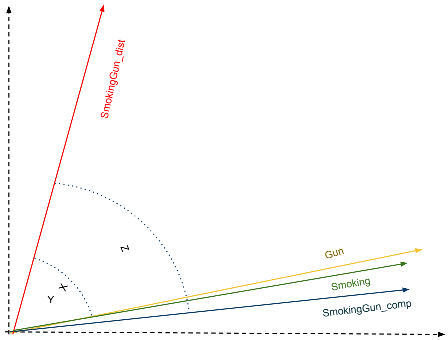
Semi-Compositional: Class3



- if Z > Y and Z < X, then multi-word is semi-compositional i.e. Class3

THE UNIVERSITY of York

Non-Compositional: Class4



- if Z > Y and Z > X, then multi-word is non-compositional i.e. Class4

Weights: A and B

- $TrafficLight_{Comp} = A\,\textbf{Traffic} + B\,\textbf{Light}$
- Weights should be dynamic (from Conclusion1)
- $Sim1 = sim(TrafficLight_{Dist}, Traffic)$= 0.624
- $Sim2 = sim(TrafficLight_{Dist}, Light)$= 0.356

Weights: A and B

- $TrafficLight_{Comp} = A\,\mathbf{Traffic} + B\,\mathbf{Light}$
- Weights should be dynamic (from Conclusion1)
- $Sim1 = sim(TrafficLight_{Dist}, Traffic)$= 0.624
- $Sim2 = sim(TrafficLight_{Dist}, Light)$= 0.356
- $A = \frac{Sim1}{(Sim1+Sim2)}$ = 0.637
- $B = \frac{Sim2}{(Sim1+Sim2)}$ = 0.363
- Working better than static values

THE UNIVERSITY *of York*

Prototype Vs Exemplar

Prototype-based

- $sim(TrafficLight_{Dist}, Traffic) = 0.624$
- $sim(TrafficLight_{Dist}, Light) = 0.356$
- $sim(TrafficLight, TrafficLight^{Prt}_{comp}) = 0.632$
- Not a strong evidence

Exemplar-based

- Exemplar-based composition is better than Prototype-based (from Conclusion2)
- Just using 2% of exemplars of Traffic and Light
- $sim(TrafficLight, TrafficLight^{Exm}_{comp}) = 0.681$

THE UNIVERSITY of York

Prototype Vs Exemplar

Prototype-based

- $sim(CouchPotato_{Dist}, Couch)$= 0.185
- $sim(CouchPotato_{Dist}, Potato)$= 0.109
- $sim(CouchPotato, CouchPotato^{Prt}_{comp})$= 0.191

Exemplar-based

- Just using 2% of exemplars of Couch and Potato
- $sim(CouchPotato, CouchPotato^{Exm}_{comp})$= 0.046

Prototype Vs Exemplar

Prototype-based

- $sim(RoastPotato_{Dist}, Roast) = 0.788$
- $sim(RoastPotato_{Dist}, Potato) = 0.462$
- $sim(RoastPotato, RoastPotato^{Prt}_{comp}) = 0.836$

Exemplar-based

- Just using 2% of exemplars of Roast and Potato
- $sim(CouchPotato, CouchPotato^{Exm}_{comp}) = 0.826$

Conclusion 3 and 4

Relative Threshold with **exemplar-based** modelling
better for compositionality detection

Summary

I presented

- Static weights don't work. Dynamic weights are better
- Exemplar-based composition vs Prototype-based composition
- Four different classes of compositionality
- Importance of relative threshold
- A method for estimating weights
- Exemplar-based model for compositionality detection

THE UNIVERSITY of York

Suggestions/Questions?
Thank You

THE UNIVERSITY of York

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.

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.

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.

Bibliography II

Guevara, E. (2010). A Regression Model of Adjective-Noun Compositionality in Distributional Semantics. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, pages 33–37, Uppsala, Sweden. Association for Computational Linguistics.

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

Lin, D. (1999). Automatic identification of non-compositional phrases. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 317–324, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mitchell, J. and Lapata, M. (2008). Vector-based Models of Semantic Composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio. 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.

THE UNIVERSITY of York

Bibliography IV

Widdows, D. (2008). Semantic vector products: Some initial investigations. In *Proceedings of the Second AAAI Symposium on Quantum Interaction*. AAAI.

Zanzotto, F. M., Korkontzelos, I., Fallucchi, F., and Manandhar, S. (2010). Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 1263–1271, Beijing, China. Coling 2010 Organizing Committee.