Using Semantics of the Arguments for Predicate Sense Induction

Anna Rumshisky
Victor A. Grinberg

September 18, 2009
GL2009 – Pisa, Italy

Resolving Lexical Ambiguity

- Words are disambiguated in context
- Our focus here will be primarily on verbs
  - though we have applied some of the same principles to noun contexts
- For verbs, main sources of sense discrimination
  - Syntactic frames
  - Semantics of the arguments

Word Sense Determined in Context

- Argument Structure (Syntactic Frame)

  The authorities denied that there is an alternative. [that-CLAUSE]

  The authorities denied the Prime Minister the visa. [NP] [NP]

- Semantic Typing of Arguments, Adjuncts, Adverbials

  The general fired four lieutenant-colonels. *(dismiss)*

  The general fired four rounds. *(shoot)*

  This development explains their strategy. *(be the reason for)*

  This booklet explains their strategy. *(describe)*

  Peter treated Mary with antibiotics. *(medical)*

  Peter treated Mary with respect. *(human relations)*

  The customer will absorb the cost. *(pay)*

  The customer will absorb this information. *(learn)*

Our Focus

- Problem addressed

  Sense distinctions linked to argument semantics

  - The customer will absorb the cost.
  - The customer will absorb this information.

- Automated algorithm for detecting such distinctions

Talk Outline

- Problem Definition
  - Resolution of Lexical Ambiguity in Verbs
  - Using Semantics of the Arguments for Disambiguation
- Review of Distributional Similarity Approaches
- Bipartite Contextualized Clustering
- Performance in Sense Induction Task
- Conclusion

Sense Induction with Argument Sets

- Sense induction based on semantic properties of the words with which the target word forms syntactic dependencies
  - will use the term selector for dependents and headwords alike
- Need to group together selectors that pick same sense of the target word

Corpus Patterns for "absorb"

The customer will absorb the cost.
Mr. Clinton wanted energy producers to absorb the tax.
PATTERN 1: [[Abstract] | [Person]] absorb [[Asset]]

They quietly absorbed this new information.
Meanwhile, I absorbed a fair amount of management skills.
PATTERN 2: [[Person]] absorb {([QUANT]) [[Abstract= Concept]]}

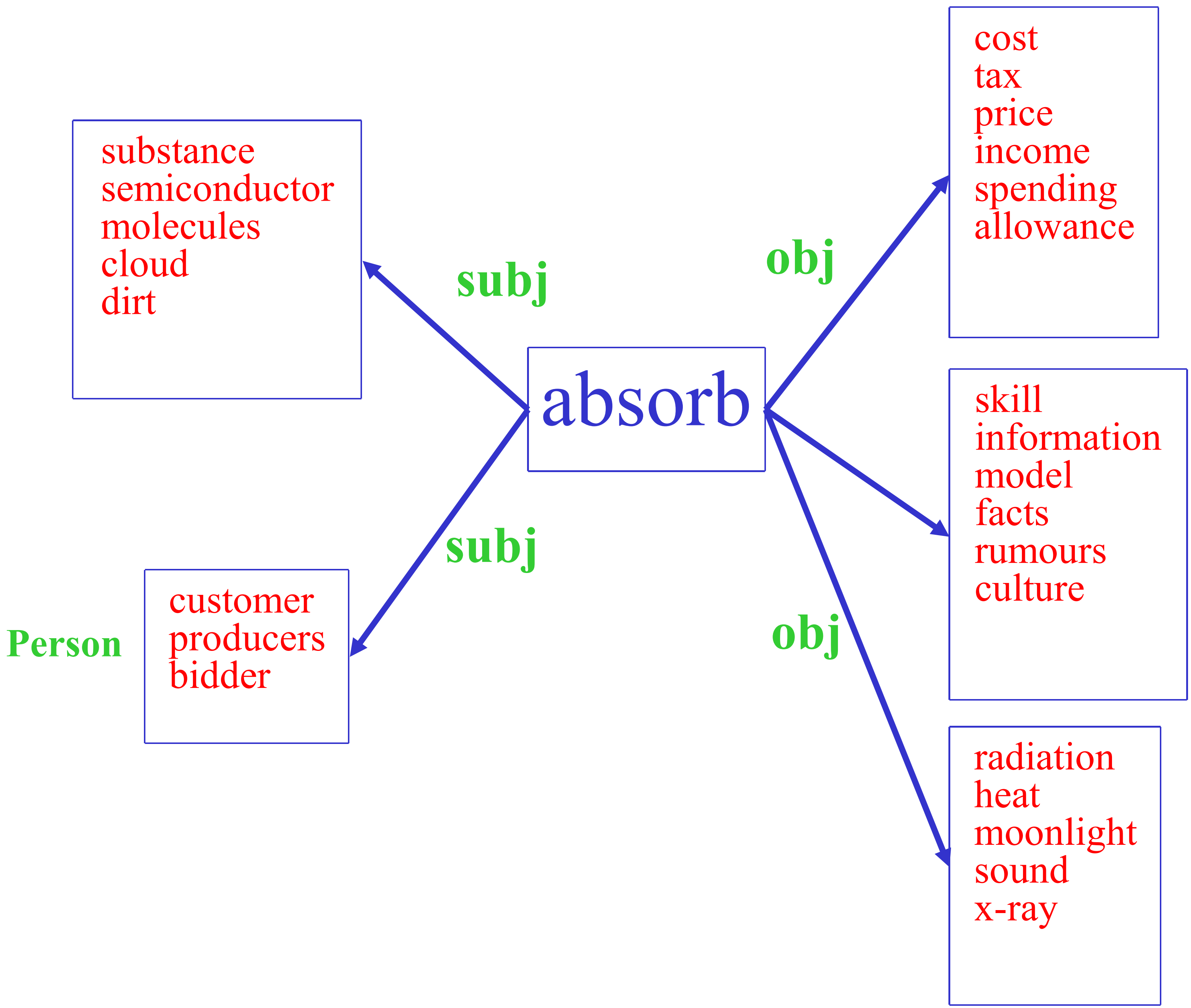
Water easily absorbs heat.
The SO_2 cloud absorbs solar radiation.
PATTERN 3: [[PhysObj] | [Substance]] absorb [[Energy]]

The villagers were far too absorbed in their own affairs.
He became completely absorbed in struggling for survival.
PATTERN 4: [[Person]] {be | become} absorbed {in [[Activity] | [Abstract]]}

---

* Patterns taken from the CPA project pattern set

Argument Sets for Different Senses



Sense Induction with Argument Sets

- Selection works in both directions with polysemous verbs
  - context elements select a particular sense of the target word
  - a given sense selects for particular aspects of meaning in its arguments
- Argument sets are often semantically heterogeneous

  absorb the {skill, information, rumours, culture}
- Running example

  deny-v (Sense 1 refuse to give / Sense 2 state that something is untrue)

  object

  a. Sense 1: visa, access, consent, approval, allowance

  b. Sense 2: accusation, rumour, charge, attack, sale, existence, presence

Distributional Similarity

- Typically, such tasks are addressed using distributional similarity
  - Get all the contexts in which the word occurs
  - Compare contexts for different words
- Context gets represented as a feature vector

  $<(feature_i, value_i)> = <(feature_1, value_1), (feature_2, value_2), \ldots>$

- Each feature corresponds to some element or parameter of the context
  - bag of words; populated grammatical relations
- Measure how close two words (e.g. skill-n, culture-n) are distributionally
  - e.g. cosine between vectors; other measures of how often words occur in similar contexts
- Measure how close two contexts of occurrence are, using distributional information on words comprising each context

Similarity Measures

$$Dice(A,B) = \frac{|A \cap B|}{\frac{1}{2}(|A|+|B|)}; \quad Dice^{\dagger}(\vec{X},\vec{Y}) = \frac{\sum_i \min(x_i, y_i)}{\frac{1}{2}\left(\sum_i x_i + \sum_i y_i\right)}$$

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|}; \quad Jaccard^{\dagger}(\vec{X},\vec{Y}) = \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)}$$

$$\cos(\bar{X},\bar{Y}) = \frac{\vec{X} \cdot \vec{Y}}{|\vec{X}||\vec{Y}|} = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$$

$$Euclidean\text{-}Distance(\vec{X},\vec{Y}) = |\vec{X} - \vec{Y}| = \sqrt{\sum_i (x_i - y_i)^2}$$

$$L_1 \; norm = \sum_i |x_i - y_i| = 2\left(1 - \sum_i \min(x_i, y_i)\right)$$

$$D(p||q) = \sum_i p_i \log \frac{p_i}{q_i}$$

$$JS(p||q) = \frac{1}{2}\left[D(p||\tfrac{p+q}{2}) + D(q||\tfrac{p+q}{2})\right]$$

$$\alpha\text{-}skew(p,q) = D(p||\alpha \cdot q + (1-\alpha) \cdot p)$$

Uses for Distributional Similarity

- Distributional similarity measures are used to produce clusters of semantically similar words
  - reciprocal nearest neighbours (Grefenstette 1994)
- Multiple senses for each word can be represented by soft cluster assignments
  - committees (Pantel & Lin 2002)
  - Sketch Engine position clusters (Kilgarriff & Rychly 2004)

Distributional Similarity

- Why can't we use it?
  - In our task, selector contexts do not need to be distributionally similar
  - They only need to be <u>similar in context</u> (= activate the same sense)

deny-v (Sense 1 refuse to give / Sense 2 state that something is untrue)

object

a. Sense 1: visa, access, consent, approval, allowance

b. Sense 2: accusation, rumour, charge, attack, sale, existence, presence

- <u>Overall distributional similarity</u> may be low

sim(visa-n, allowance-n); *sim*(sale-n, rumour-n)

- But <u>contextualized similarity</u> must be high

c_sim(visa-n, allowance-n, (deny-v, object))

What we propose

- A method to contextualize distributional representation of lexical items to a particular context
- Sense induction technique based on this contextualized representation

Talk Outline

- Problem Definition
  - Resolution of Lexical Ambiguity in Verbs
  - Using Semantics of the Arguments for Disambiguation
- Review of Distributional Similarity Approaches
- Bipartite Contextualized Clustering
- Performance in Sense Induction Task
- Conclusion

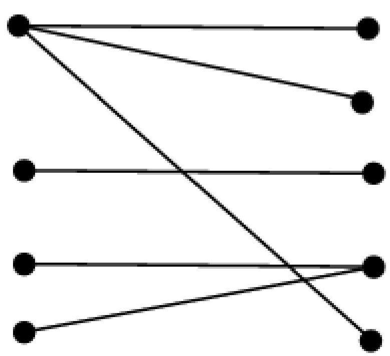
Bipartite Contextualized Clustering

Bipartite Contextualized Clustering

- Each sense of the target word selects for a particular semantic component
- Identifying selectors that activate a given sense of the target is equivalent to identifying other contexts that select for the same semantic component
  - Therefore, must cluster words that select for the same properties as a given sense of the target – with respect to the target word and a particular grammatical relation: e.g., (acquire, object)
- acquire (learn vs. buy):

hone	skill
practice	language
master	technique
learn	habit
...	...
purchase	land
own	stock
sell	business
steal	property
...	...

Think about it as a bipartite graph:



Selectional Equivalence

- A word is a *selectional equivalent* of the target word if one of its senses, selects (in the specified argument position) for the same meaning component as one of the senses of the targer word

    acquire

    - (*purchase*): purchase, own, sell, buy, steal
        - land, stock, business
    - (*acquire a quality*): emphasize, stress, recognize, possess, lack
        - significance, importance, meaning, character
    - (*learn*): hone, practice, teach, learn, master
        - skill, language, technique

- Selectional equivalents for a given sense of the target word occur with the same selectors as that sense and effectively ensure that we perceive that selector as activating that sense of the target
    - land and stocks can be purchased and owned, skills and techniques can be practiced and taught, hence we acquire them in a different sense

Procedure (1)

- Identify potential selectional equivalents for different senses of the target
  - Identify all selector contexts in which the target word was found in corpus.
    - (selector, gramrel): e.g., (stock, $object^{-1}$)
  - Take the inverse image of the above set under grammatical R^{-1}. This gives a set of potential equivalents for each sense of the target.

Procedure (2)

- Identify relevant selectors, i.e. good disambiguators that activate similar interpretations for the target and its potential equivalent
  - Given the target word *t* and potential selectional equivalent *w*
    - Compute association scores for each selector *s* that occurs with both *t* and *w*
    - Combine the two association scores using a combiner function $\psi(assoc_R(s, t), assoc_R(s, w))$
    - Choose top-*k* selectors that maximize it!
      - Each potential selectional equivalent is represented as a *k*-dimensional vector $w = \langle f(s) \rangle$ of resulting selector scores

How do we do it?
(identify relevant selectors)

Given the target (deny-v, object):

- for confirm-v, we would need to select report-n, existence-n, allegation-n
- for grant-v, we would need to select access-n, right-n, approval-n, permission-n

Relevant selectors must occur “often enough” with both words

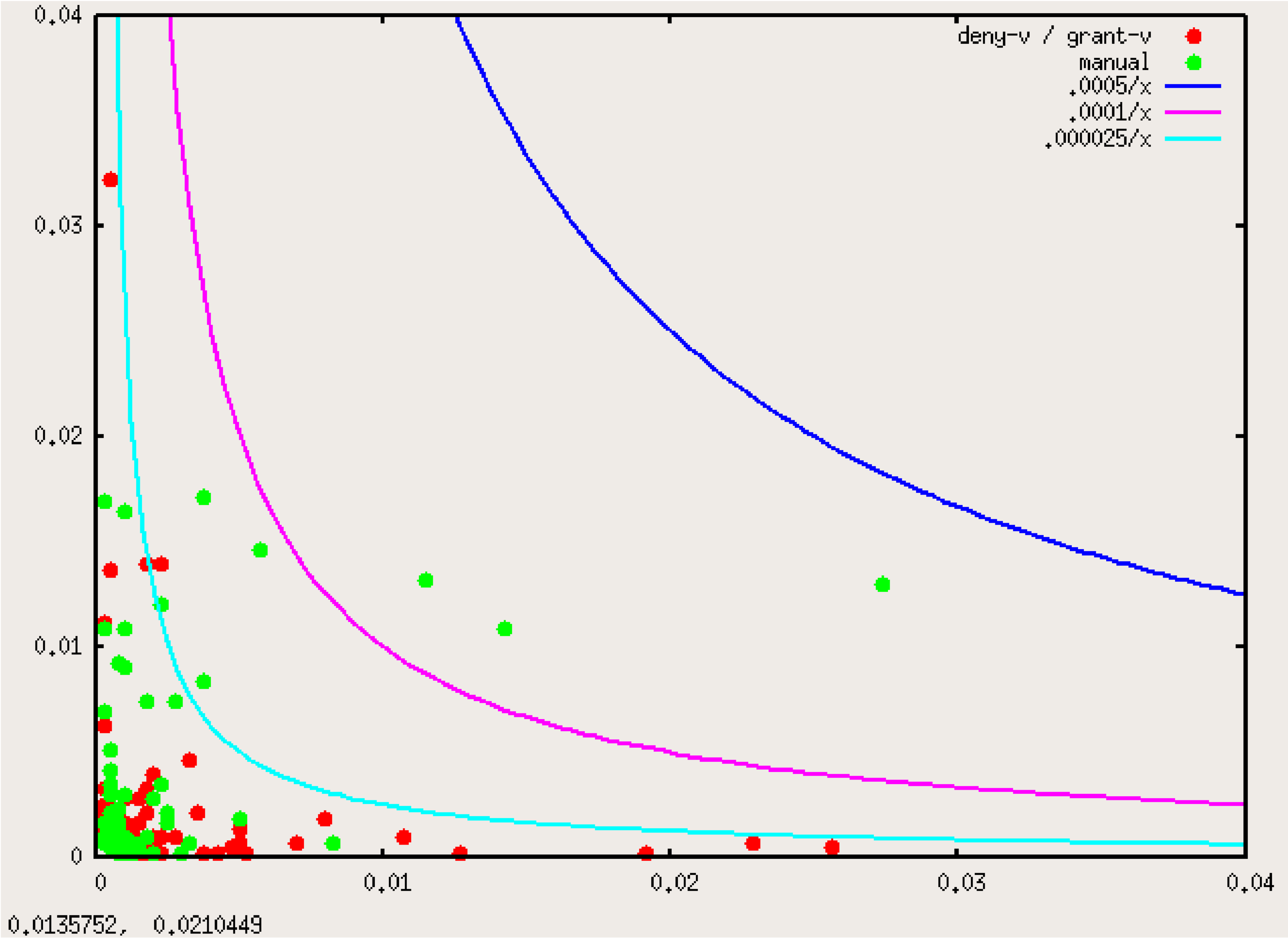
- modeled as having both association scores relatively high

System Configurations

- Association scores for (selector, verb, relation)
  - $P(s|Rw)$
  - $mi(s,Rw)$
  - $mi(s,Rw)$ * log freq(s, R, w)
- Combiner functions $\psi(assoc_R(s, t), assoc_R(s, w))$
  - product $a_1 a_2$ ← equivalence classes along hyperbolic curves
  - harmonic mean $2a_1 a_2/(a_1+a_2)$

Choosing selectors for deny-v/grant-v

(with R = object)



Identifying Reliable Selectors

Assoc. score: Conditional probability

	deny-v		confirm-v	
	count	P(n|Rv)	count	P(n|Rv)
'report-n'	103	.0256	62	.0159
'existence-n'	92	.0228	32	.0082
'claim-n'	77	.0191	17	.0043
'allegation-n'	99	.0246	7	.0018
'view-n'	8	.0019	86	.0221
'importance-n'	32	.0079	18	.0046
'fact-n'	20	.0049	23	.0059
'involvement-n'	63	.0156	6	.0015
'charge-n'	184	.0457	2	.0005
'right-n'	57	.0141	6	.0015

Identifying Reliable Selectors

Assoc. score: Conditional probability

	deny-v		grant-v	
	count	P(n|Rv)	count	P(n|Rv)
'access-n'	110	.0273	56	.0129
'right-n'	57	.0141	46	.0108
'approval-n'	46	.0114	57	.0132
'permission-n'	9	.0022	228	.0528
'rights-n'	23	.0057	63	.0145
'status-n'	15	.0037	74	.0171
'charge-n'	184	.0457	5	.0011
'power-n'	9	.0022	60	.0139
'request-n'	15	.0037	36	.0083
'license-n'	2	.0049	254	.0588

Resulting Representations

Assoc. score: Conditional probability

	confirm-v	grant-v	refuse-v
	P(s|Rw)	P(s|Rw)	P(s|Rw)
'access'	.0000	.0129	.0145
'rights'	.0015	.0108	.0017
'approval'	.0005	.0132	.0009
'permission'	.0000	.0528	.0660

	confirm-v	contradict-v	refuse-v
	P(s|Rw)	P(s|Rw)	P(s|Rw)
'report'	.0160	.0108	.0000
'story'	.0039	.0054	.0000
'allegation'	.0018	.0027	.0000
'view'	.0222	.0376	.0000

Identifying Reliable Selectors

Assoc. score: Mutual information

	deny-v		confirm-v	
	count	MI(n,Rv)	count	MI(n,Rv)
'ascendency-n'	1	11.6	2	12.8
'appropriateness-n'	3	9.1	2	8.7
'validity-n'	17	8.9	10	8.3
'centrality-n'	1	7.7	3	9.5
'primacy-n'	2	7.6	5	9.1
'existence-n'	83	8.9	76	7.4
'rumour-n'	28	9.1	7	7.3
'sighting-n'	1	7.0	3	8.8
'prejudice-n'	6	7.3	9	8.0
'allegation-n'	91	10.7	2	5.4

Identifying Reliable Selectors

<u>Assoc. score: Mutual information</u>

	deny-v		grant-v	
	count	MI(n,Rv)	count	MI(n,Rv)
'approval-n'	37	8.5	21	8.6
'serf-n'	1	7.4	3	9.9
'primacy-n'	2	7.6	3	9.1
'visa-n'	2	7.1	5	9.3
'permission-n'	4	5.6	71	10.6
'autonomy-n'	5	6.7	8	8.3
'access-n'	48	7.5	23	7.3
'exemption-n'	1	5.0	28	10.6
'request-n'	11	6.3	21	8.1
'asylum-n'	1	5.3	8	9.1

Choosing Association Scores

- Conditional probability gives equal weight to each instance, regardless of how frequent the selector itself is
- MI scheme picks more "characteristic", but less frequent selectors
  - Normalizing for selector frequency,
  - Intersection between selector lists is smaller, similarity computation becomes unreliable
- Normalizing MI by the log factor de-emphasizes selectors with low occurrence counts

Procedure (3)

- Produce clusters of selectional equivalents
  - *group-average agglomerative clustering*
  - similarity measure:
    - sum of minima of association scores (numeric equivalent of set intersect)
  - intra- and inter-cluster APS
    - average pairwise similarity is kept both for elements within each cluster, and for every pair of merged clusters
  - ranked selector lists
    - keep a list of selectors for each node in the cluster tree
    - a union of selector lists computed, each selector assigned the score equal to the weighted average of its scores in the merged clusters
    - soft cluster assignment for selectors

Merging Ranked Selector Lists

– selector lists for (acquire, object)

```
Cluster 2234=564+667 (45.351/45.351) [stress-v] [underline-v]
<pre-eminence-n:8.91 distinctiveness-n:8.77 significance-n:7.47 credentials-n:7.26
importance-n:7.20 dimension-n:6.99 salience-n:5.15 respectability-n:4.17 reputation-n:3.83
humility-n:3.72 individuality-n:3.70 liturgy-n:3.66 normality-n:3.57 urgency-n:3.43 liking-n:3.39
gloss-n:3.37 fascination-n:3.33 status-n:3.29 elegance-n:3.29 hollow-n:3.25 competence-n:3.25
orientation-n:3.24 sensitivity-n:3.16 willingness-n:3.14>

Cluster 747 [emphasise-v]
<distinctiveness-n:8.55 stigma-n:8.25 longevity-n:8.11 tan-n:7.89 individuality-n:7.66
credentials-n:7.66 legitimacy-n:7.32 significance-n:7.25 importance-n:7.17 hollow-n:6.94
reputation-n:6.81 attribute-n:6.69 trait-n:6.50 relevance-n:6.41 status-n:6.32>

Cluster 2239=747+2234 (43.648/44.215) [emphasise-v] [stress-v underline-v]
<distinctiveness-n:8.70 significance-n:7.40 credentials-n:7.39 importance-n:7.19
pre-eminence-n:5.94 individuality-n:5.02 reputation-n:4.82 dimension-n:4.66 hollow-n:4.48
status-n:4.30 salience-n:3.43 respectability-n:2.78 stigma-n:2.75 longevity-n:2.70 tan-n:2.63
humility-n:2.48 legitimacy-n:2.44 liturgy-n:2.44 normality-n:2.38 urgency-n:2.29 liking-n:2.26
gloss-n:2.24 attribute-n:2.23 fascination-n:2.22 elegance-n:2.19 trait-n:2.17 competence-n:2.16
orientation-n:2.16 relevance-n:2.14 sensitivity-n:2.11 willingness-n:2.09>
```

Implementation

- Custom-designed agglomerative clustering engine implemented in C++
  - easy extension for different scoring schemes, similarity measures, hard/soft clustering schemes
- 100M word British National Corpus
- Robust Accurate Statistical Parsing (RASP) used to extract grammatical relations
  - binary relations (dobj, subj, etc.)
  - ternary relations (w/ introducing preposition)
    - frequency-filtered (e.g. rare prepositions thrown out)
  - relation inverses for all relations

Talk Outline

- Problem Definition
  - Resolution of Lexical Ambiguity in Verbs
  - Using Semantics of the Arguments for Disambiguation
- Review of Distributional Similarity Approaches
- Bipartite Contextualized Clustering
- Performance in Sense Induction Task
- Conclusion

Sense-Induction Task

Sense Induction Task

- We adapted our system for use in a standard word sense induction (WSI) setting
- Recent Semeval-2007 (Agirre et al. 2007) competition had a WSI task in which 6 systems competed
- 65 verbs were used in the data set
  - unsuitable for our purposes, as sense distinctions due to argument semantics impossible to identify
  - a lot of verbs with senses that depend for disambiguation on syntactic frame
- We use a separately developed data set and perform comparison relative to the baselines

Data Set Characteristics

- We needed a data set that targets a specific contextual factor
  - namely, the semantics of a particular argument
- 15 (verb, grammatical relation) pairs
  - verbs judged to have sense distinctions dependent on a particular argument (we chose **dobj**)
- 200 instances for each pair; two annotators
- Inter-annotator agreement (ITA) 95% micro-average
  - range 99% – 84%
- Average number of senses 3.65 (range: 2-11, stddev: 2.30)

Data Set, Per-word Characteristics

- Distribution across senses
  - Per-verb entropy much higher than for SemEval data
- Tested in supervised learning setting
  - MaxEnt accuracy

Word	No. Senses	No. Inst.	ITA %	Entropy	MFS	MaxEnt accuracy
absorb	7	196	92.4	2.49	.30	.58
acquire	4	186	92.1	1.86	.44	.44
admit	2	163	98.7	1.00	.53	.71
assume	3	191	90.8	1.55	.45	.73
conclude	2	178	97.5	0.96	.62	.89
cut	4	166	92.3	1.33	.58	.51
deny	3	190	97.2	1.49	.49	.62
dictate	2	193	98.9	0.53	.88	.97
drive	11	174	97.6	2.64	.41	.40
edit	2	176	98.0	0.98	.57	.82
enjoy	2	193	86.2	0.93	.66	.70
fire	6	162	97.3	1.87	.54	.73
grasp	3	178	97.6	1.25	.49	.84
know	2	172	92.6	0.98	.58	.79
launch	3	196	89.9	1.24	.63	.74
Average	3.73	180.9	94.5	1.41	.545	.699

Sketch Engine

Home | Concordance | Word Sketch | Thesaurus | Sketch-Diff

dictate **bnc freq = 1264**

object	646	4.2	subject	520	6.8	modifier	175	2.2
letter	25	20.51	circumstance	14	18.39	otherwise	11	25.83
pace	12	19.87	consideration	10	15.82	partly	9	24.2
term	26	19.1	custom	7	15.19	largely	7	18.48
choice	14	16.57	prudence	3	14.77	also	14	16.25
policy	21	15.33	tradition	7	12.64	often	8	15.4
shape	9	13.71	sense	10	12.49	externally	2	13.01
caution	4	12.51	conscience	4	12.41	still	7	12.6
intifada	2	10.95	availability	4	11.56	only	7	11.89
content	5	10.51	logic	4	11.47	clearly	4	11.89
form	10	9.15	wisdom	3	9.93	necessarily	3	11.49
tactic	3	9.08	arithmetic	2	9.22	probably	4	11.37
action	8	9.03	Quinn	2	8.85	even	5	11.04
passage	4	8.86	paradigm	2	8.44	merely	3	10.96
memoirs	2	8.86	function	5	8.36	partially	2	10.15
geometry	2	8.67	convenience	2	8.28	already	4	9.83
pattern	6	8.4	fashion	3	7.92	ultimately	2	9.81
format	3	8.25	Brussels	2	7.87	always	4	9.26
need	7	8.14	need	6	7.8	virtually	2	8.89
treatment	5	7.84	interest	7	7.78	effectively	2	8.45
answer	4	7.64	policy	7	7.67	both	3	8.27

Senses for dictate, dobj

(1) verbalize to be recorded (letter, passage, memoir)

(2) determine the character of or serve as a motivation for (terms, policy, etc.)

Using Clusters in a WSI Task

(1) Sort all the nodes in the dendrogram by computing rank of each node C_i

$$rank(C_i) = IntraAPS(C_i) \cdot \log(|C_i|) \cdot \log(\sum_{s \in C_i} f_i(s))$$

(2) Given selector *s* from a particular corpus occurrence of target, compute an association score for each of the chosen clusters C_i and *s*

$$\text{assoc}(s, C_i) = \frac{\sum_{w \in C_i} \text{mi}(s, Rw)}{|C_i|}$$

where

$$\text{mi}(s, Rw) = \log \frac{P(s,R,w)}{P(s)P(R,w)}$$

Using Clusters in a WSI Task

(3) For each sentence in the data set, we extract the selector which in that sentence occurs in the specified grammatical relation to the target

(4) For each of the extracted selectors,

- selector-cluster association score is computed with each of the top-ranking clusters in the dendrogram
- sentences containing that selector are associated with the highest-ranking cluster

(5) Sentences associated with intersecting verb clusters (i.e. clusters containing at least some of the same selectional equivalents of the target) are grouped together

Evaluation Measures

1. Set-matching F-measure (Agirre et al., 2007)
   - computer F-measure for each cluster/sense class pair
   - find the optimal cluster for each sense
   - average F-measure of the best-matching cluster across all senses
2. Harmonic mean of B-Cubed P&R (Amigo et al, 2008)
   - based on per-element Precision and Recall

$$\text{BCubed Precision} = \frac{\sum_e \frac{|c_e \cap s_e|}{|c_e|}}{n}$$

$$\text{BCubed Recall} = \frac{\sum_e \frac{|c_e \cap s_e|}{|s_e|}}{n}$$

where e is an element of data set D, c_e is the cluster to which e belongs, s_e is the sense class to which e belongs, and $n = |D|$

Evaluation Measures (2)

3. <u>NormalizedMI</u>

- We define mutual information *I(C,S)* between the two variables defined by the clustering solution *C* and the gold standard sense assignment *S* as

$$I(C,S) = \sum_{i,j} P(i,j) \log \frac{P(i,j)}{P(i)P(j)}$$

– where c_i is a cluster from C, s_i is a sense from S, and $P(i,j) = \frac{|c_i \cap s_j|}{n}$

(similar to Meila 2003)

- Range for the mutual information depends on the entropy values *H(C)* and *H(S)*

$$0 \leq I(C,S) \leq \min(H(C), H(S))$$

Evaluation Measures (3)

3. NormalizedMI (cont'd)
   - We normalize this value by max(*H(C)*,*H(S)*)

$$\text{NormalizedMI} = \frac{I(C,S)}{\max(H(S),H(C))}$$

   - This normalization gives us some desirable properties for comparison across data sets
     i. (0,1) range
     ii. NormalizedMI*(1c1word, S)* = 0
     iii. NormalizedMI*(1c1inst, S)* = *H(S)*/log *n*

Baselines

We used the same the baselines as in the SEMEVAL WSI Task

- 1 cluster 1 word
  - all occurrences are grouped together for each target word
- 1 cluster 1 instance
  - each instance is a cluster (i.e. singleton)

Senseval System Performance

System	F-measure		BCubed		Norm. MI	
		% 1c1w		% 1c1w		% 1c1i
1c1inst	.035	4.6	.039	5.0	.118	100
1c1word	.755	100	.776	100	0	0
I2R	.528	69.9	.505	65.1	.051	**43.2**
UBC-AS	.750	**99.3**	.769	**99.1**	.005	4.2
UMND2	.640	84.8	.638	82.2	.006	5.1
UOY	.383	50.7	.253	32.6	.048	40.7
upv_si	.607	80.4	.520	67.0	.044	37.3

Our System Performance

System	F-measure		BCubed		Norm. MI	
		% 1c1w		% 1c1w		% 1c1i
1c1inst	.038	6.5	.040	6.7	.188	100
1c1word	.584	100	.599	100	0	0
mi-fact-prod	.586	**100.3**	.522	87.1	.138	**73.4**
mi-fact-prod-prod	.572	97.9	.540	**90.2**	.061	32.4
mi-prod	.504	86.3	.439	73.3	.103	54.8
mi-prod-prod	.544	93.2	.469	78.3	.101	53.7

- Results reported for configurations selected in preliminary evaluation

System-Specific Considerations

- This method has an obvious disadvantage, compared to the full WSI systems
  - disambiguation is based on a single selector
- The system performs well despite this handicap

- The verbs in our data set have sense distinctions that depend on the semantics of the chosen argument
  - this disadvantage should manifest only in cases where other context elements contribute to disambiguation

Talk Outline

- Problem Definition
  - Resolution of Lexical Ambiguity in Verbs
  - Using Semantics of the Arguments for Disambiguation
- Review of Distributional Similarity Approaches
- Bipartite Contextualized Clustering
- Performance in Sense Induction Task
- Conclusions

Conclusions

- A method to contextualize distributional representation of lexical items to a particular context
- Resulting clustering algorithm produces groups of words selectionally similar to different senses of the target, with respect to the specified argument position
- Fully unsupervised
- Avoid computational pittfalls by using short contextualized vectors

Practical Applications

- Enhance lexicographic analysis and research tools that facilitate sense definition (e.g. the Sketch Engine, Kilgarriff & Rychly 2004)
- Should help improve performance of complete WSD or WSI systems, possibly facilitate various parsing tasks, counteract data sparsity problem in a number of tasks

Thank you!

Overlapping Senses

- Frequently, there are good prototypical cases that exemplify each sense

  The research showed an undeniable dependency

  The photo showed the victim entering the store

- And then there are boundary cases

  The graph showed an undeniable dependency

Per-word System Performance

Word	No. Senses	No. Inst.	ITA %	Entropy	MFS	MaxEnt accuracy	F-measure		
							random	1c1word	mi-fact-prod
absorb	7	196	92.4	2.49	.30	.58	.20	.33	.36
acquire	4	186	92.1	1.86	.44	.44	.30	.45	.59
admit	2	163	98.7	1.00	.53	.71	.51	.67	.74
assume	3	191	90.8	1.55	.45	.73	.39	.52	.48
conclude	2	178	97.5	0.96	.62	.89	.55	.68	.51
cut	4	166	92.3	1.33	.58	.51	.49	.61	.78
deny	3	190	97.2	1.49	.49	.62	.38	.54	.55
dictate	2	193	98.9	0.53	.88	.97	.79	.85	.62
drive	11	174	97.6	2.64	.41	.40	.23	.34	.39
edit	2	176	98.0	0.98	.57	.82	.57	.67	.62
enjoy	2	193	86.2	0.93	.66	.70	.57	.70	.53
fire	6	162	97.3	1.87	.54	.73	.37	.49	.58
grasp	3	178	97.6	1.25	.49	.84	.45	.61	.85
know	2	172	92.6	0.98	.58	.79	.54	.67	.56
launch	3	196	89.9	1.24	.63	.74	.52	.62	.66
Average	3.73	180.9	94.5	1.41	.545	.699	.457	.584	.586