Unsupervised Role Labelling

Summer 2012: Seminar
Recent Developments in Computational Semantics

Sibel Ciddi
sciddi@coli.uni-saarland.de

Outline

- Definition of Semantic Roles
- Unsupervised Semantic Role Induction, *Joel Lang and Mirella Lapata, 2011*
  - Split-Merge Clustering & Graph Partitioning
    - \- Argument Identification
    - \- Argument Classification
- A Bayesian Model for Unsupervised Semantic Parsing, *Ivan Titov and Alexandre Klementiev (2011)*
  - Argument & Predicate Identification
  - Semantic Role Label Assignment for arguments

What are Semantic Roles?

According to Gildea and Jurafsky:

```
Semantic relations->  “who” did “what” to “whom”,“when”,
“where”, and “how”.
```

Example: predicate *break*

Jim broke the window with a ball.

Semantic Roles:

PROTO-AGENT (A0) – Jim $_{A0}$ broke the window with a ball *(who?)*

PROTO-PATIENT (A1) - Jim $_{A0}$ broke the window $_{A1}$ with a ball *(to whom / what?)*

INSTRUMENT (A3) – Jim $_{A0}$ broke the window $_{A1}$ with a ball $_{A3}$ *(with what?)*

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Previous related works on role labeling:

Supervised and Semi-supervised works:

- Using syntactically similar (labeled) structures for PropBank, Gordon & Swanson (2007)
- Single-source language projected annotation *(previously labeled),* Fuerstenau & Lapata, 2009
- Multi-language projected annotation, Pad and Lapata (2009).

Unsupervised works:

- [1] Bootstrapping using VerbNet, Swier and Stevenson (2004)
- [2] Directed graphical model via Latent variables: Grenager & Manning (2006)
- Predicate argument algorithm based on aPoS annotations Abend et al. (2009)
- Core and adjunct roles, using an unsupervised parser and PoS-tagger, Abend and Rapport (2010)
- [3] Role induction based on logistic classifier with hidden variables for canonical syntactic forms, Lang & Lapata, (2010)

Motivation:

- Argument identification & labelling → "classification“
  → Supervised learning

- Dependence on large annotated data sets on current systems
  - Annotation work → expensive & time-consuming
  - low coverage and domain dependent

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Why are Unsupervised methods good?

CoNLL 2008 benchmark dataset on SRL

- ✓ Does not compromise F1
- ✓ Provides better cluster purity
- ✓ Computationally efficient
- ✓ Does not rely on hidden variables

Multiple Outputs

- o training materials for supervised learning
- o information extraction
- o question answering
- o machine translation
- o summarization

How?

Treat argument identification and role labeling separately.

<u>Split & Merging Method</u>

- Predicate argument identification
  → *via linguistic rules*
- Assign semantic roles
  →*via Split & Merge of clusters*

<u>Graph Partitioning Method</u>

- Predicate argument identification
  → *Syntactic processing on dependency tree*
- Assign semantic roles
  → *via Argument Classification to make partitioning graphs*

Learning Setting

Dependency-based representation:

- Simplified argument identification, and cluster assignment
- Consistency with the CoNLL 2008 benchmark data used for evaluation

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Argument Identification (*decision for semantic-role induction*):

- Discard non-semantic argument candidates
- Select semantic argument candidates

  ****In Graph Partitioning model, the final decision for the positive identification of an argument is made in the classification step.*

Identification Cues?

- PoS and the syntactic relations
  - From predicate to argument on the dependency tree
- For each candidate, the first matching rule applies.

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Argument Identification Rules, *For each candidate:*

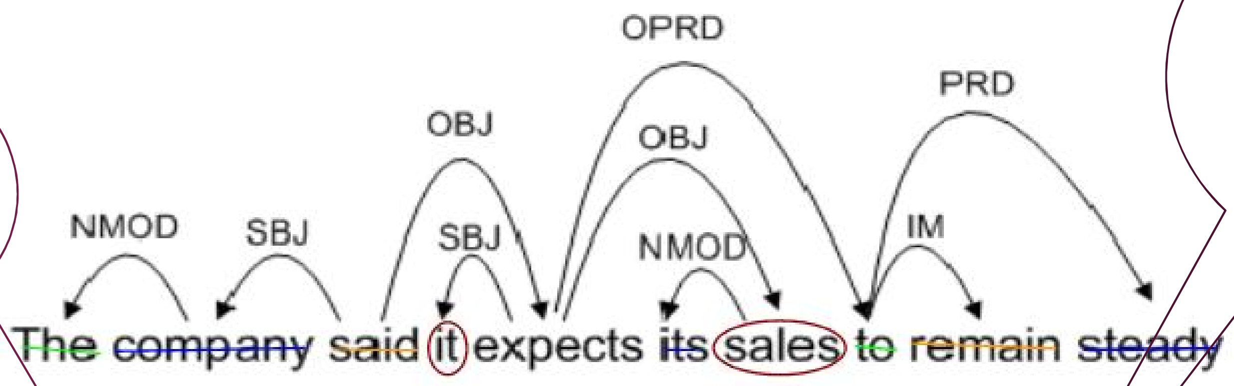
1. **Discard**: Determiner, Infinitive Marker, Coordinative Conjunction, punctuation.
2. **Discard:** the path relations from predicate to candidate ending with coordinate, subordinate.
3. **Keep:** The closest subject to the left of a predicate **and** the relations from predicate p to the governor g—*that are all upward-leading (g →p)*
4. **Discard:** the path between the predicate and the candidate except for the last relation containing a subject relation, adjective modifier relation, etc…
5. **Discard:** The auxiliary verbs.
6. **Keep:** the parent predicates *(of the candidate)*.
7. **Keep:** the path from predicate to candidate leading to several verbal nodes, and ending with an arbitrary relation.
8. **Discard** the rest.

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Argument Identification:

Example:

Discard it if it is an auxiliary verb....



Keep it if it is the closest subject to the left of a predicate and the candidate relations from *p*. to *g*. are all upward-leading (g-->p)

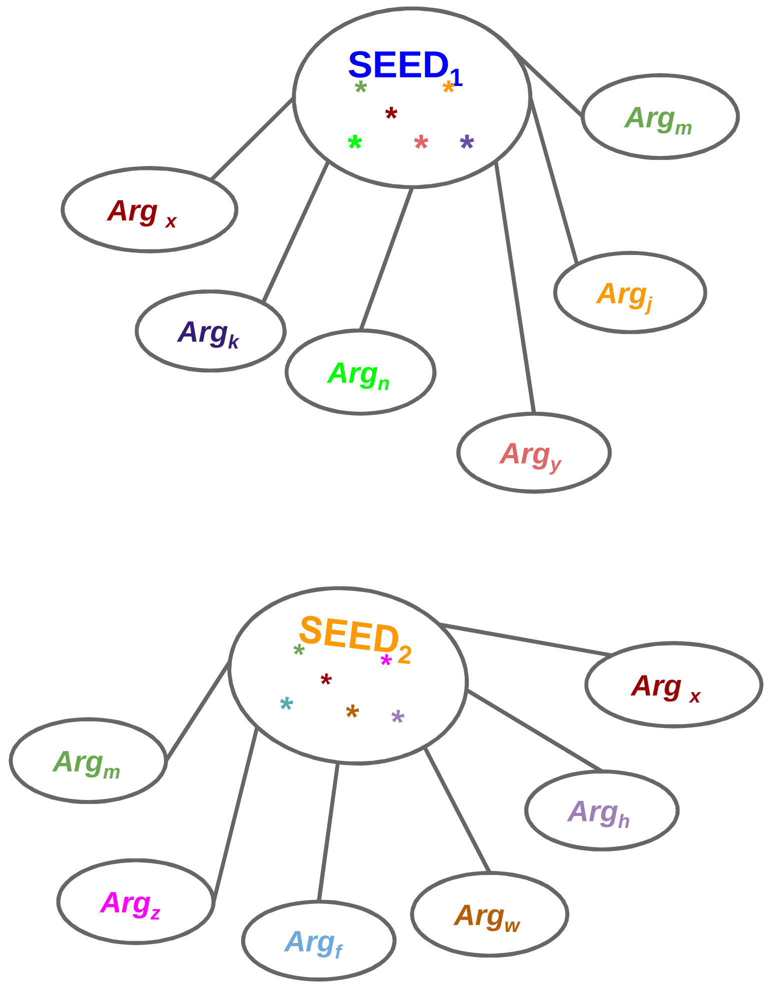
1. Discard "**the**" and "**to**" due to PoS (rule (1)).
2. Discard "**remain**" because the path ends with inf. marker rel.; discard "**said**" because the path ends with an upward-leading OBJ relation (rule (2)).
3. ~~Rule (3) does not match and is therefore not applied.~~
4. Discard "**steady**" because there is a downward-leading O-PRD relation along the path; and discard "**company**" and "**its**" because of the OBJ relations along the path (rule (4)).
5. ~~Rule (5) does not apply.~~
6. Keep the words "**it**" and "**sales**" as likely arguments (rule (6)).
7. ~~There are no more candidates left, so the Rule (7) does not apply.~~

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Split & Merge

1. **Split-Merge Role Induction:**

- assigning argument instances to cluster to represent sem. roles
  - separate clusters for each verb => verb-specific role (*according to Propbank)*
- *Iterative splitting & merging clusters*
  - *High-purity but low collocation*
  - *Degree of dislocation : Consecutive Merge*



Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Graph Partitioning

Argument Classification

By using an undirected, weighted graph for each verb where:

- Vertices correspond to verb argument instances
- Edge weights quantify the similarities between the argument instances
    - Partition the argument-instance graph into clusters of vertices representing semantic roles and each argument instance with a label (which indicates its associated cluster).

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Graph Partitioning

Graph Construction (*undirected, weighted graph)*

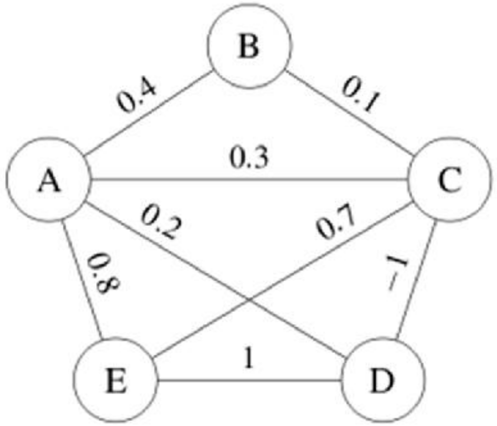


Figure 1: Simplified example of an argument-instance

G = (V, E, φ) (vertices, edges, weight function)

→ Add each verb argument instance as a vertex.

→ For each possible pair of vertices (vi , v j),

→ compute weight:

→ If the result is non-zero:

→ Add an edge for the vertices of (vi, vj) with weight to the graph.

the similarity between instances *(values from φ function):*

→ (+): the roles are likely to be the same

→ (-): the roles are likely to be different

→ If the values are zero, neither one of the roles are likely.

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Split & **Merge** *a. Split Phase* | b. *Merge Phase*

i. **Candidate Search:** *Considering only large* clusters:*

- Specific clusters
- Certain other (*larger*) clusters
  - Merging scores between them

ii. **Scoring Function:** *Similarity score of clusters*

- Lexical similarity: $lex(x,y) = cossim(x,y) = \frac{x \cdot y}{||x|| \, ||y||}$
- Clause-level constraints: $cons(c,c') = 1 - \frac{2 * viol(c,c')}{NC + NC'}$
- PoS similarity *(cosine-similarity):*

  Clusters → vectors x, y = (arg-PoS, occurrence frequency values).

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Graph Partitioning

ii. Similarity score of vertex labels

1. If each vertex v_i gets a label l_i ,all the instances of v_i go into cluster l_i
2. The label for each vertex gets iteratively updated.

*Each label gets a score that is a sum of the weights of edges in nearest (neighboring) vertices with that label and select the label with the maximum score.***

Example: Updating vertex A.

Vertex B, E: same label (*the same cluster)*
Vertex C, D: in different clusters. If:

- score for cluster {B, E} = 0.4 + 0.8 = 1.2
- score for cluster {C}= 0.3
- score for cluster {D}= 0.2.

Vertex A= assigned to cluster {B, E} → the highest score.

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Split & Merge *a. Split Phase* | b. *Merge Phase*

Note on Scoring Function *Thresholds*

- Negative Evidence:
  - Clause-level constraint threshold: γ =0
  - PoS similarity: β = 0
    - They can rule out incorrect merges, but can't identify correct ones.
    - Two clusters with different PoS unlikely to represent same role.

- PoS similarity does **NOT** imply role-semantic similarity.
- If (pos) and (cons) similarity are below $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ thresholds, then lexical similarity score assigns the final score.

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Scoring Function for Graph Partitioning-Classification *Almost same as the S&M method*

Argument-Instance Similarity

The Criteria* on determining the similarity of argument instances:

1. Are the instances lexically similar?
2. Do they occur in the same syntactic position?
3. Do they occur in the same clause?

- Lexical and syntactic similarity are determined via function:

  $\text{lex}(v_i, v_j)$, $\text{syn}(v_i, v_j)$, with range $[-1, 1]$
- The argument position according to clause (*frame*) is determined with its own function (equation (4)), which assures that two argument instances v_i and v_j occurring in the same frame cannot have the same semantic role.

iii. Threshold Adaptation & Termination:

- Start with higher values-->rule out merges via negative evidence
- Lower the threshold for a reliable merge.

Example:

β and γ value is at 0.95

- *lower β by 0.05 at each step, leaving γ unchanged*
- *When β is 0, lower γ by 0.05*
- *Reset β to 0.95.*
- *After γ is lowered by another 0.05*
- *Decrease β iteratively until it is 0.*

**** The algorithm terminates, when γ is 0 after repeating these steps.*

*** **Graph Partitioning Method** applied the same technique.*

Experimental Set-up

Data:

- Evaluation: CoNLL-2008 Shared Task Data Set *(PropBank-style gold standard annotations)*.

- The training: the WSJ part of the Penn Treebank, converted into ***a dependency format***, and automatic parses obtained from the MaltParser.

    - Used only verbal predicate-argument constructions annotations.

Evaluation Metrics & Comparison Models:

1. **Purity:** *the extent of argument instances in the clusters which share the same gold standard role*

$$PU = \frac{1}{N} \sum_i \max_j |G_j \cap C_i|$$

2. **Collocation:** *the extent to which a particular gold standard role is assigned to a single cluster*

$$CO = \frac{1}{N} \sum_j \max_i |G_j \cap C_i|$$

3. **Clustering Quality - F_1:** *the harmonic mean of purity and collocation as a single measure of clustering quality, obtain F_1*

$$F_1 = \frac{2 \cdot CO \cdot PU}{CO + PU}$$

Evaluation Metrics & Comparison Models:

	Syntactic Function			Latent Logistic			Split-Merge			Graph Partitioning		
	PU	CO	F1	PU	CO	F1	PU	CO	F1	PU	CO	F1
auto/auto	72.9	73.9	73.4	73.2	**76.0**	74.6	81.9	71.2	**76.2**	**82.5**	68.8	75.0
gold/auto	77.7	**80.1**	**78.9**	75.6	79.4	77.4	**84.0**	74.4	**78.9**	**84.0**	73.5	78.4
auto/gold	77.0	71.0	73.9	77.9	**74.4**	76.2	86.5	69.8	**77.3**	**87.4**	65.9	75.2
gold/gold	81.6	**77.5**	79.5	79.5	76.5	78.0	**88.7**	73.0	**80.1**	88.6	70.7	78.6

Table 1: Evaluation of the output of our graph partitioning algorithm compared to our previous models and a baseline that assigns arguments to clusters based on their syntactic function.

- The Graph Partitioning better on F1 than Syn.Func. auto/auto and auto/gold
  - (-) ~0.5 on the gold/auto, and (-)~0.9 points on the gold/gold dataset.
- Highest purity on all datasets
  - (-) except for gold/gold, and - 0.1 less than Split-Merge.

Unsupervised Semantic Role Induction, Lang & Lapata, 2011

Split & Merge Results:

		Syntactic Function			Split-Merge		
Verb	Freq	PU	CO	F1	PU	CO	F1
say	15238	91.4	91.3	**91.4**	93.6	81.7	87.2
make	4250	68.6	71.9	70.2	73.3	72.9	**73.1**
go	2109	45.1	56.0	49.9	52.7	51.9	**52.3**
increase	1392	59.7	68.4	63.7	68.8	71.4	**70.1**
know	983	62.4	72.7	**67.1**	63.7	65.9	64.8
tell	911	61.9	76.8	68.6	77.5	70.8	**74.0**
consider	753	63.5	65.6	64.5	79.2	61.6	**69.3**
acquire	704	75.9	79.7	77.7	80.1	76.6	**78.3**
meet	574	76.7	76.0	76.3	88.0	69.7	**77.8**
send	506	69.6	63.8	66.6	83.6	65.8	**73.6**
open	482	63.1	73.4	67.9	77.6	62.2	**69.1**
break	246	53.7	58.9	56.2	68.7	53.3	**60.0**

*Showing results for 12 verbs, for varied occurrence frequencies and alternation patterns.

The increase in F1 and purity — also holds across verbs.

Conclusion:

- No human manual effort for training.
- With the rule-based automatic argument candidate identification, no supervision.
- Benefits for supervised methods:
  - Automatic annotation with manual correction
  - Providing useful out-of-domain data for training.

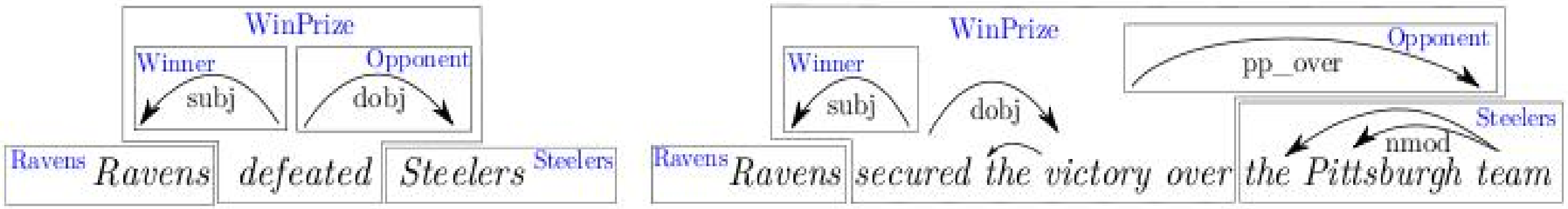
For Results:

- Graph Partitioning can get higher purity than others by trading off collocation, better F1
- As for the Split & Merge method, if the clusters induced by the S&M system re-annotated, low collocation requires in higher annotation effort
  - However, low purity results in poorer data quality.

A Bayesian Model for Unsupervised Semantic Parsing, Titov and Klementiev, 2011

What is Semantic Parsing & Representation ?

Recursive prediction of predicate argument structure, and clustering of argument fillers.



Tasks for identification of frames and arguments:

1. Divide the sentence into lexical items
2. Assign cluster labels over lex. items
   -predicates get a "semantic frame"
   -arguments get a "argument filler cluster"
3. Determine argument relations between lex. items

Related Work

supervised role labeling:

- Gildea and Jurafsky, 2002
- Johansson, 2008

unsupervised RL:

- Lang and Lapata, 2010, 2011: Split & Merge
- Grenager and Manning, 2006

A Bayesian Model for Unsupervised Semantic Parsing

Description

Statistical dependencies between predicates and arguments, by means of Inference algorithm from a modified version of Metropolitan-Hastings split-merge sampler.

Motivation

Same motivations that are valid for previous unsupervised role labeling are also valid for this one.

Using distributional context in the Bayesian model, for unsupervised role labeling

-The model significantly outperforms previous works that are more intensive

A Bayesian Model for Unsupervised Semantic Parsing

Markov Logic Networks (MLNs)

Based on Poon and Domingo's (2009)

- dependency tree joint probability, and its latent semantics
    - via Markov Logic Network *(Richardson and Domingos, 2006)*
    - selecting ***parameters***
- Maximizes the probability of the observed dependency structures.

For <u>Semantic Parsing,</u>

- They modified the parameters and local normalization constraints

For <u>Bayesian Model</u> based Semantic Parsing,

- This work focuses on non-parametric Bayesian techniques.

A Bayesian Model for Unsupervised Semantic Parsing

Non-parametric version of Bayesian model, as part of the Dirichlet processes *(Ferguson, 1973).*

Pitman-Yor processes model the following:

i. distributions of semantic classes appearing as an argument of other semantic classes.
ii. distributions of syntactic realizations for each semantic class
iii. distributions of syntactic dependency arcs for argument types.

How is SRL done by Bayesian Model?

- Predicate arguments identification:

  → According to argument occurrences with argument keys (signatures)

- Semantic role assignment (labeling):

  → By clustering argument keys *(Lang & Lapata, 2011)*

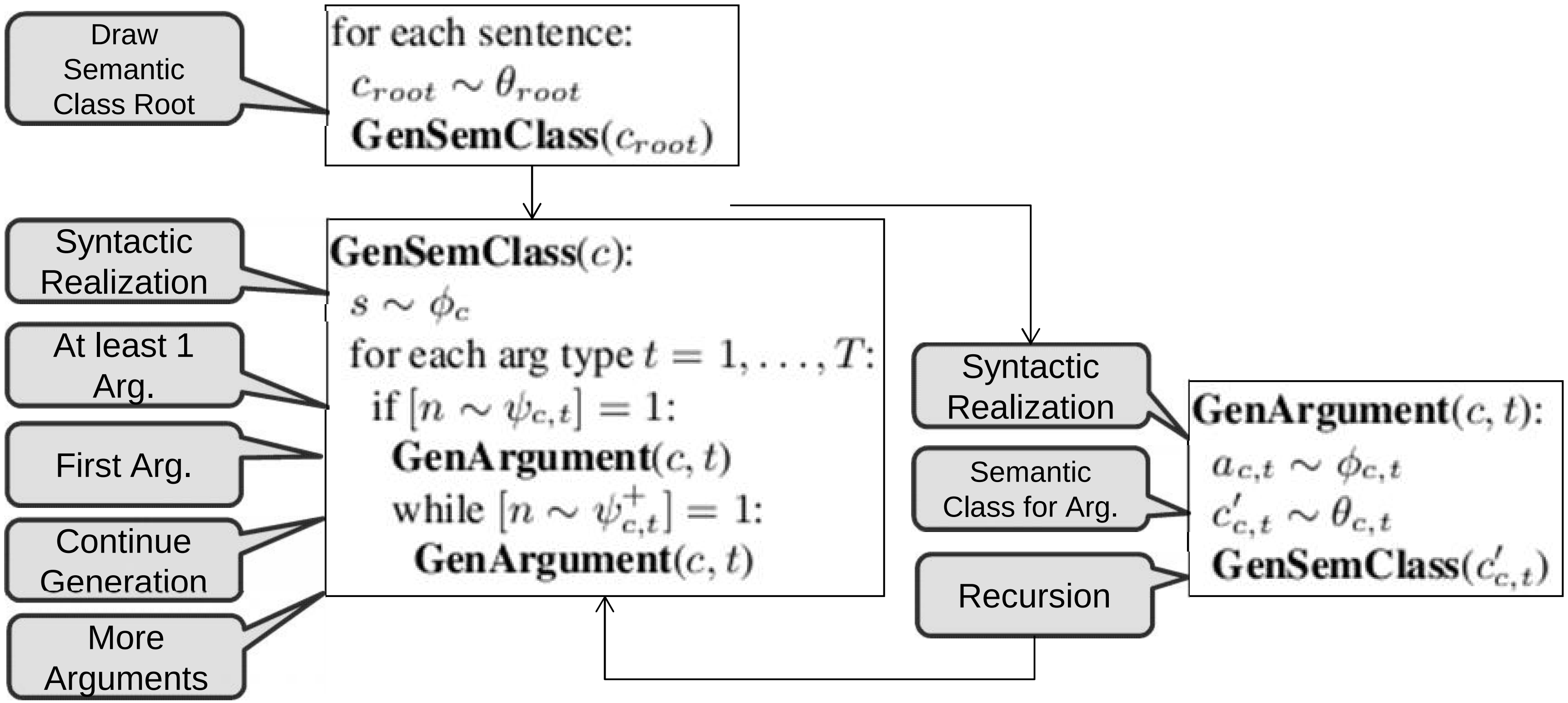
Main Goal

- Induce semantic roles automatically from unannotated texts

  → Assuming sentences have auto-annotated syntactic trees *(Equivalent to clustering of argument occurrences)*

A Bayesian Model for Unsupervised Semantic Parsing

Model for Semantic Parsing



Model Parameters (from Hierarchical Pitman-Yor Processes)

- The distribution of syntactic and lexical items for each semantic class *(evokes frame elements for each frame)*

- The distribution over mixed argument roles for each semantic class

- Linking between semantic argument roles and syntactic dependencies for each semantic class

- Selectional preferences for each argument type: a distribution over semantic classes of argument fillers

Inference:

Estimated using sentences automatically annotated with syntactic structures and treating their semantic representation as latent Metropolis-Hastings moves.

i. Split and Merge
ii. Compose - Decompose
iii. Role syntax alignment
iv. Informed Proposals

A Bayesian Model for Unsupervised Semantic Parsing

Inference Steps

Split-Merge: Two semantics classes merged together, or one class decomposed into two

Compose – Decompose: Two lexical items decomposed, or one lexical item decomposed: "*secure*" and "*victory*" to obtain "secure – → victory".

Role syntax alignment: Mapping between semantic roles, and syntactic fragments, by **Using a greedy randomized search "Gibbs Scan".**

Informed Proposals:

- Split-Merge move based on cosine similarity of lexical and syntactic contexts
- Compose-Decompose based on the connections of fragment pairs' frequency

A Bayesian Model for Unsupervised Semantic Parsing

EVALUATION *(From GENIA Corpus for Q/A Task)*

Class	Variations
1	motif, sequence, regulatory element, response element, element, dna sequence
2	donor, individual, subject
3	important, essential, critical
4	dose, concentration
5	activation, transcriptional activation, transactivation
6	b cell, t lymphocyte, thymocyte, b lymphocyte, t cell, t-cell line, human lymphocyte, t-lymphocyte
7	indicate, reveal, document, suggest, demonstrate
8	augment, abolish, inhibit, convert, cause, abrogate, modulate, block, decrease, reduce, diminish, suppress, up-regulate, impair, reverse, enhance
9	confirm, assess, examine, study, evaluate, test, resolve, determine, investigate
10	nf-kappab, nf-kappa b, nfkappab, nf-kb
11	antiserum, antibody, monoclonal antibody, ab, antisera, mab
12	tnfalpha, tnf-alpha, il-6, tnf

Table 1: Examples of the induced semantic classes.

A Bayesian Model for Unsupervised Semantic Parsing

EVALUATION

	Total	Correct	Accuracy
KW	150	67	45%
KW-SYN	87	67	77%
TR-EXACT	29	23	79%
TR-SUB	152	81	53%
RS-EXACT	53	24	45%
RS-SUB	196	81	41%
DIRT	159	94	59%
USP-MLN	334	295	88%
USP-BAYES	325	259	80%

Table 2: Performance on the QA task.

Question: *What does cyclosporin A suppress?*

Answer: *expression of EGR-2*

Sentence*: As with EGR-3 , expression of EGR-2 was blocked by cyclosporin A .*

Question*: What inhibits tnf-alpha?*

Answer: *IL -10*

Sentence: *Our previous studies in human monocytes have demonstrated that interleukin (IL) -10 inhibits lipopolysaccharide (LPS) -stimulated production of inflammatory cytokines , IL-1 beta , IL-6 , IL-8 , and tumor necrosis factor (TNF) -alpha by blocking gene transcription .*

CONCLUSION

- More than half of the mistakes are due to over-coarse clustering in 3 semantic classes (6, 8, 12)
- The current work reflects results only on Genia corpus for Q/A Task
  - For Comparative Analysis, using CoNNL data sets?

Unsupervised Role Labelling

Questions & Comments ?