Georgia Maniati
University of Edinburgh

March 8, 2016

Unsupervised Induction of Semantic Roles

—

Joel Lang and Mirella Lapata (2010)

Overview

Introduction

Semantic Roles

The Semantic Role Labeling Task (SRL)

Problem Formulation

Role induction as a clustering problem

Standard linking & alternations

Model

Extension of logistic classifier with latent variables

Evaluation

Summary

Introduction

The Semantic Role Labeling task

Semantic roles: labels that capture aspects of the semantics of the relationship between predicate and argument while abstracting over surface syntactic configurations

- ➢ Predicate - Argument
- ➢ Agent - Patient

[Michael]$_{\textit{Agent}}$ **eats** [a sandwich]$_{\textit{Patient}}$.

[A sandwich]$_{\textit{Patient}}$ is **eaten** by [Michael]$_{\textit{Agent}}$.

- ❖ Common Role Annotation Frameworks:
  - FrameNet: frame-specific roles
  - **PropBank:** Proto-roles

Introduction

Contingency table between *syntactic function* and *semantic role* for two core roles and two adjunct roles (counts from CoNLL 2008).

- ➢ **84.5%** of **A0** (Proto-Agent) roles are **subjects**
- ➢ **58.4%** of **A1** (Proto-Patient) roles are **objects**

★ **Linking theory** assumption- tendency of semantic role be mapped onto single syntactic function

PropBank

	A0	A1	TMP	MNR
SBJ	54514	19684	15	7
OBJ	3359	51730	93	54
ADV	162	3506	976	2308
TMP	5	60	15167	22
PMOD	2466	4860	142	62
OPRD	37	5554	1	36
LOC	17	145	43	157
DIR	0	178	15	6
MNR	5	48	13	3312
PRP	9	50	11	6
LGS	2168	36	2	2
PRD	413	830	31	38
NMOD	422	388	25	59
EXT	0	20	2	12
DEP	18	150	25	65
SUB	3	84	4	2
CONJ	198	331	22	8
ROOT	62	147	84	2
	64517	88616	16803	6404

Introductio

The Semantic Role Labeling task

Goal: automatically classify the arguments of a predicate with semantic roles

Full SRL system:

- ➢ predicate identification
- ➢ argument identification
- ➢ **argument classification**

Challenge: computational treatment of syntactic alternations

Introduction

Supervised approaches:

- ➢ parse the training corpus
- ➢ match **labeled semantic roles** to syntactic functions
- ➢ extract features from the parse tree
- ➢ train a probabilistic model on the features

But

Hand-labeled data are **domain & language specific** and **expensive** to produce .

Solution: mechanism for inducing the semantic roles from **unlabeled data**

Problem Formulation

Argument classification as a **clustering problem**:

- ➢ A set of clusters for each predicate (predicate specific PropBank roles)
- ➢ Each cluster corresponds to a semantic role
- ➢ Ideally one-to-one mapping between each cluster and each semantic role

Reformulated task :

- ➢ assign the arguments of a specific predicate to one of the clusters associated with it

Problem Formulation How to deal with **syntactic alternations**?

Each predicate is associated with a **standard linking**: the most frequent mapping of the *syntactic function* of its arguments to *semantic role*s.

[Michael]$_{A0}$ **eats** [a sandwich]$_{A1}$.

- ➢ standard linking for predicate ‘to eat’:
  - ○ Subject-A0
  - ○ Object-A1

Canonical function: the syntactic function an argument would have had, if the standard linking had been used.

[A sandwich]$_{Patient}$ is **eaten** by [Michael]$_{Agent}$.

- ➢ canonical function for argument ‘A sandwich’: Object

Problem Formulation

Sub-problems

1) **Detection** of non-standard linkings
2) **Canonicalization**: determine canonical function
3) **Clustering** according to canonical function

Sub-problems 1 & 2 rely on the distribution *p(F)* over the possible canonical functions *F* of an argument.

3) For each predicate we have K clusters:

Order syntactic functions by occurrence frequency.

- For each of the K-1 most frequent functions allocate a separate cluster.
- Assign all remaining functions to the Kth cluster.

Model

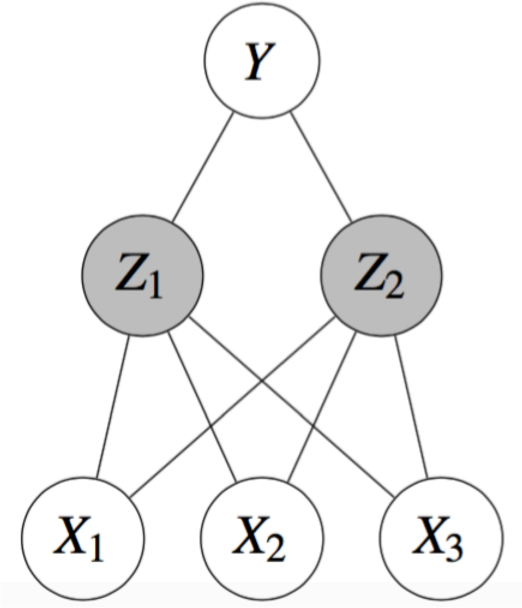
How do we estimate *p(F)* ?

- ➢ Extension of logistic classifier with latent variables to avoid overfitting

Goal: learn the canonical function of arguments for each predicate

Training data: parser output - most observed syntactic functions will correspond to canonical functions

Features: at or below node representing argument head in parse tree



The logistic classifier with latent variables illustrated as a graphical model in unrolled form for M=2 and N=3.

X_1, X_2, X_3: observed features
Z_1, Z_2: binary latent variables
Y: observed target

Model

How do we estimate p(F) ?

$$p(y,z|x,\theta) = \frac{1}{P(x,\theta)} \exp\left(\sum_k \theta_k \phi_k(x,y,z)\right)$$

- probability distribution over the target variable *Y* and the latent variables *Z,* conditional on the input variables *X*
- each of the feature functions *φ* is associated with a parameter *θ*

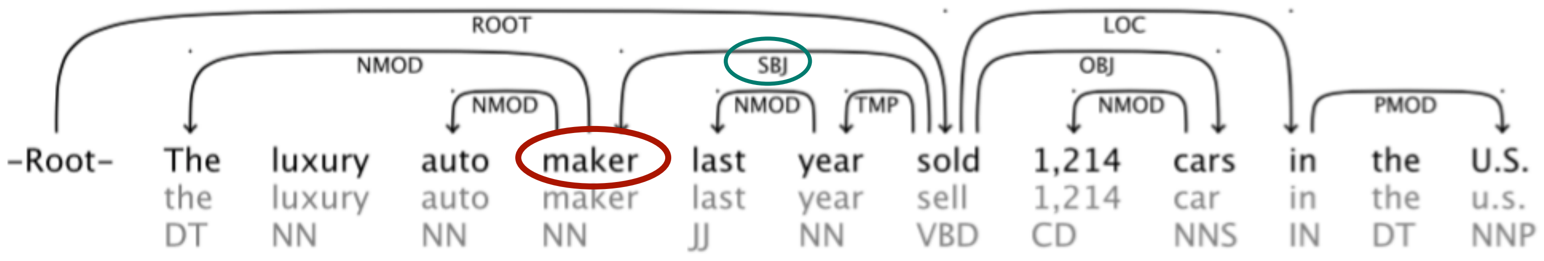
$$\begin{aligned} l(\theta) &= \log p(d|c) \\ &= \sum_i \log \sum_z p(d_i, z|c_i) \\ &= \sum_i \log \frac{\sum_z \exp(\sum_k \theta_k \phi_k(c_i,d_i,z))}{P(c_i,\theta)} \end{aligned}$$

$$\downarrow$$

$$\begin{aligned} (\nabla l)_k &= \frac{\partial}{\partial \theta_k} l(\theta) \\ &= \sum_i \sum_z p(z|d_i,c_i)\phi_k(c_i,d_i,z) \\ &\quad - \sum_i \sum_{y,z} p(y,z|c_i)\phi_k(c_i,y,z) \end{aligned}$$

- For a training set of inputs *c* and corresponding targets *d*, we obtain the maximum-likelihood parameters by finding the *θ* maximizing *l(θ)*

Model



Dependency graph of a sample sentence from the corpus

Features extracted:
predicate lemma, argument lemma, argument POS, preposition involved (if any), lemma of left-most/right-most child of the argument, POS of left-most/right-most child of argument, a key formed by concatenating all syntactic functions of the argument's children

- ➢ The features for the argument ***maker*** are:
  [*sell, maker, NN, –, the, auto, DT, NN, NMOD+NMOD*]

- ➢ The **target** for this instance (and observed syntactic function) is **SBJ**.

Evaluation

- ➢created gold standard role labeled argument instances
- ➢10 clusters for each predicate

Measures

- ➢cluster purity (PU)

$$PU = \frac{1}{K} \sum_i \max_j |c_i \cap g_j|$$

Let K denote the number of clusters, c_i the set of instances in the i-th cluster and g_j the set of instances having the j-th gold standard semantic role label.

Evaluation

Measures

- ➢cluster accuracy (CA)
- ➢cluster precision (CP)
- ➢cluster recall (CR)

$$CA = \frac{TP+TN}{TP+FP+TN+FN}$$

$$CP = \frac{TP}{TP+FP} \qquad CR = \frac{TP}{TP+FN}$$

TP : number of *pairs of instances* which have the same role and are in the same cluster,
TN : number of pairs of instances which have different roles and are in different clusters
FP : number of pairs of instances with different roles in the same cluster
FN : number of pairs of instances with the same role in different clusters

Evaluation

Performance

- ➢ better than the baseline syntactic function model
- ➢ successful in detecting alternate linkings
- ➢ higher cluster purity score compared to the Grenager and Manning’s system

Summary

- ➢ Novel framework for unsupervised role induction
- ➢ Concept: detect alternate linkings and find their canonical syntactic form
- ➢ Model:
  extends the logistic classifier with latent variables
  trained on parsed output which is used as a noisy target for learning
- ➢ Potential:
  embed argument identification system
  replace treebank trained parser with chunker

References

Lang, Joel, and Mirella Lapata. "Unsupervised induction of semantic roles." *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. Computational Linguistics, 31(1):71–106.

Georgia Maniati
University of Edinburgh
March 8, 2016

Thank you!