Natural Language Understanding

Lecture 14: Semantic Role Labeling

Adam Lopez
Slide credits: Frank Keller

March 27, 2018

School of Informatics
University of Edinburgh
keller@inf.ed.ac.uk

Reading: Zhou and Xu, 2015.

Background: Jurafsky and Martin, Ch. 22 (online 3rd edition).

Introduction

Introduction

Earlier in this course we looked at *parsing* as a fundamental task in NLP. But what is parsing actually good for?

Parsing breaks up sentences into meaningful parts or finds meaningful relationships, which can then feed into *downstream semantic tasks:*

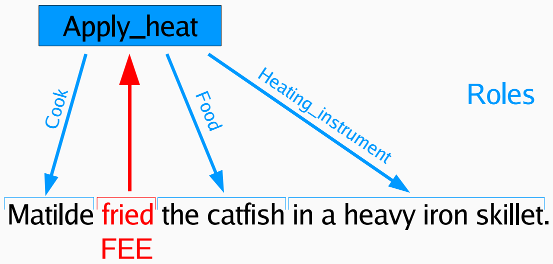
- semantic role labeling (figure out who did what do whom);
- semantic parsing (turn a sentence into a logical form);
- word sense disambiguation (figure out what the words in a sentence mean);
- compositional semantics (compute the meaning of a sentence based on the meaning of its parts).

In this lecture, we will look at *semantic role labeling* (SRL).

Introduction

Frame Semantics

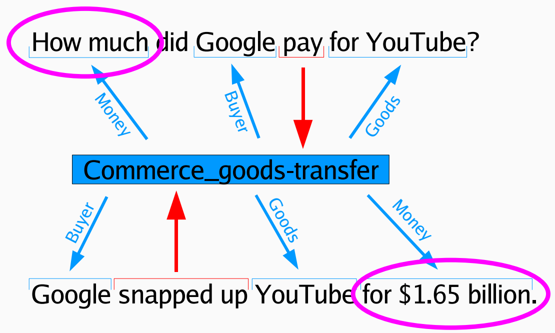
- due to Fillmore (1976);
- a frame describes a prototypical situation;
- it is evoked by a frame evoking element (predicate);
- it can have several frame elements (arguments; sem. roles).



Introduction

Properties of Frame Semantics

- provides a shallow semantic analysis (no modality, scope);
- granularity in between "universal" and "verb specific" roles;
- generalizes well across languages;
- can benefit various NLP applications (IR, QA).



Proposition Bank

PropBank is a version of the Penn Treebank annotated with semantic roles. More coarse-grained than Frame Semantics:

Propbank	Frames
Arg0	proto-agent
Arg1	proto-patient
Arg2	benefactive, instrument, attribute, end state
Arg3	start point, benefactive, instrument, or attribute
Arg4	end point
ArgM	modifier (TMP, LOC, DIR, MNR, etc.)

Arg2–Arg4 are often verb specific.

PropBank Corpus

Example (from Jurafsky and Martin):

(1) increase.01 "go up incrementally"
Arg0: causer of increase
Arg1: thing increasing
Arg2: amount increased by, EXT, or MNR
Arg3: start point
Arg4: end point

(2) [$_{\text{Arg0}}$ Big Fruit Co.] increased [$_{\text{Arg1}}$ the price of bananas].

(3) [$_{\text{Arg1}}$ The price of bananas] was increased again [$_{\text{Arg0}}$ by Big Fruit Co.]

(4) [$_{\text{Arg1}}$ The price of bananas] increased [$_{\text{Arg2}}$ 5%].

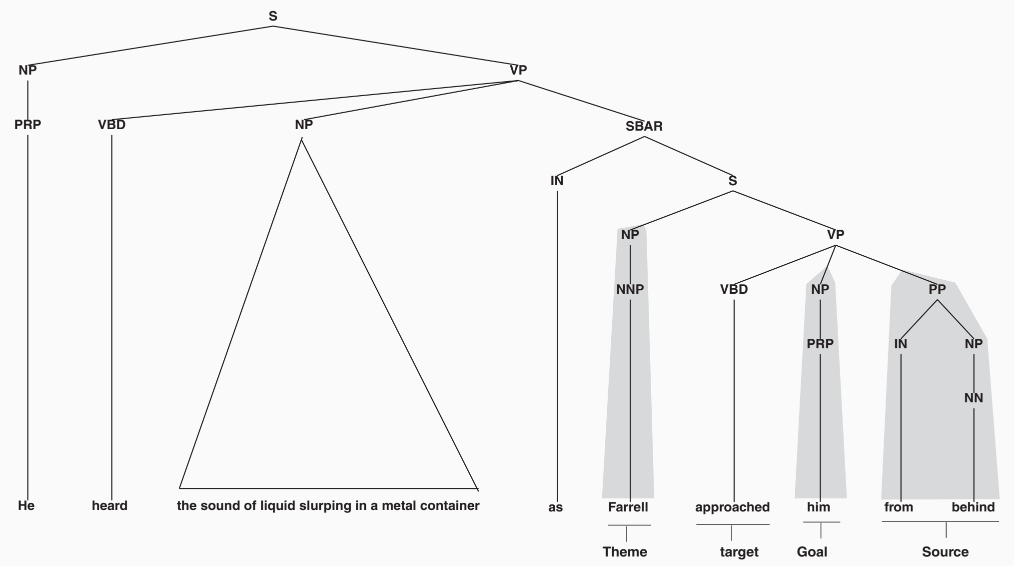
The SRL Pipeline

The SRL task is typically broken down into a sequence of sub-tasks:

1. parse the training corpus;
2. match frame elements to constituents;
3. extract features from the parse tree;
4. train a probabilistic model on the features.

More recent SRL systems use dependency parsing, but follow the same pipeline architecture.

Match Frame Elements



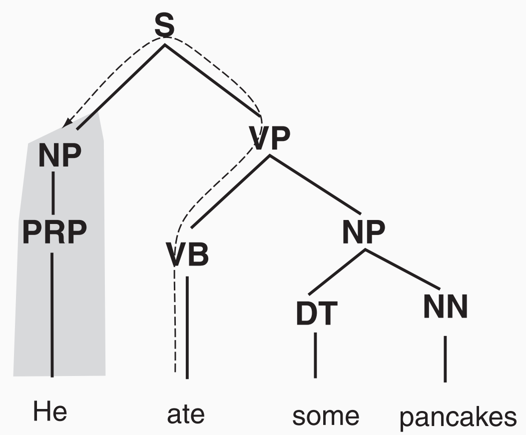
Extract Parse Features

Assume the sentences are parsed, then the following features can be extracted for role labeling:

- **Phrase Type:** syntactic type of the phrase expressing the semantic role (e.g., NP, VP, S);
- **Governing Category:** syntactic type of the phrase governing the semantic role (NP, VP), only used for NPs;
- **Parse Tree Path:** path through the parse tree from the target word to the phrase expressing the role;
- **Position:** whether the constituent occurs before or after the predicate; useful for incorrect parses;
- **Voice:** active or passive; use heuristics to identify passives;
- **Head Word:** the lexical head of the constituent.

Extract Parse Features

Path from target *ate* to frame element *He*: VB↑VP↑S↓NP



How might you do this if you had a dependency parse instead of a constituent parse?

Semantic Role Labeling with Neural Networks

Semantic Role Labeling with Neural Networks

Intuition. SRL is a sequence labeling task. We should therefore be able to use recurrent neural networks (RNNs or LSTMs) for it.

$\underbrace{\text{A record date}}_{\text{ARG1}}$ has $\underbrace{\text{n't}}_{\text{AM-NEG}}$ been **set** .

A	record	date	has	n't	been	**set**	.
B-ARG1	I-ARG1	I-ARG1	O	B-AM-NEG	O	B-V	O

Case study: SRL with deep bidirectional LSTMS

In this lecture, we will discuss the end-to-end SRL system of Zhou and Xu using a *deep bi-directional LSTM (DB-LSTM):*

Zhou and Xu approach:

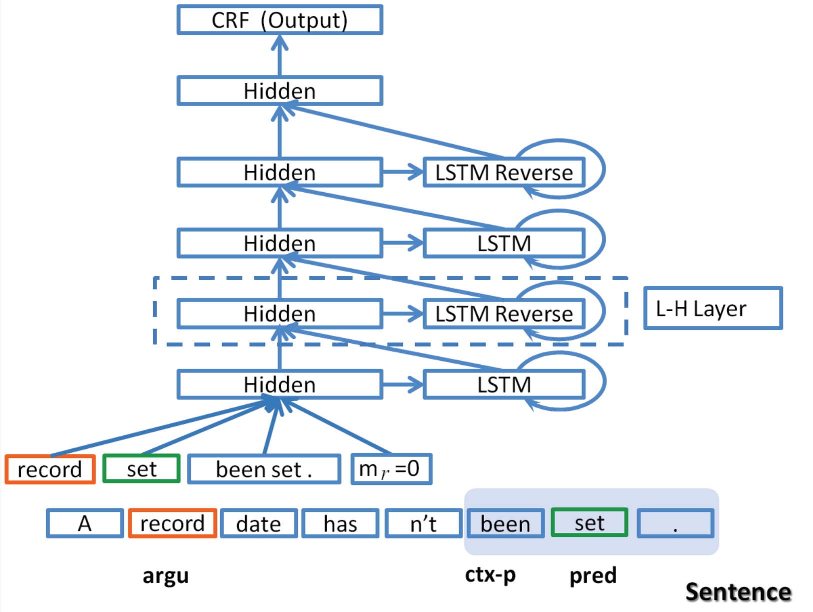
- uses no explicit syntactic information;
- requires no separate frame element matching step;
- needs no expert-designed, language-specific features;
- outperforms previous approaches using feedforward nets.

Architecture

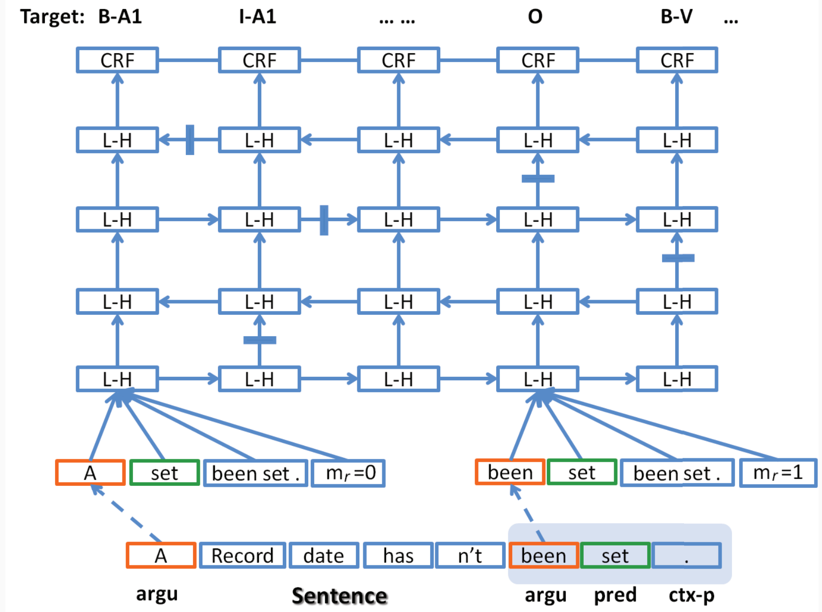
The DB-LSTM is an two-fold extension of the standard LSTM:

- a *bidirectional* LSTM normally contains two hidden layers, both connected to the same input and output layer, processing the same sequence in opposite directions;
- here, the bidirectional LSTM is used differently:
  - a standard LSTM layer processes the input in forward direction;
  - the output of this LSTM layer is the input to another LSTM layer, but in reverse direction;
- these LSTM layer pairs are stacked to obtain a deep model.

Architecture



Architecture: Unfolded



Features

The input is processed word by word. The input features are:

- argument and predicate: the argument is the word being processed, the predicate is the word it depends on;
- predicate context (ctx-p): the words around the predicate; also used to distinguish multiple instances of the same predicate;
- region mark (m_r): indicates if the argument is in the predicate context region or not;
- if a sequence has n_p predicates it is processed n_p times.

Output: semantic role label for the predicate/argument pair using IOB tags (inside, outside, beginning).

Features

An example sequence with the four input features: argument, predicate, predicate context (ctx-p), region mark (m_r):

Time	Argument	Predicate	ctx-p	m_r	Label
1	A	set	been set .	0	B-A1
2	record	set	been set .	0	I-A1
3	date	set	been set .	0	I-A1
4	has	set	been set .	0	O
5	n't	set	been set .	0	B-AM-NEG
6	been	set	been set .	1	O
7	set	set	been set .	1	B-V
8	.	set	been set .	1	O

Training

- Word embeddings are used as input, not raw words;
- the embeddings for arguments, predicate, and ctx-p, as well as m_r are concatenated and used as input for the DB-LSTM;
- eight bidirectional layers are used;
- the output is passed through a conditional random field (CRF); allows to model dependencies between output labels;
- the model is trained with standard backprop using stochastic gradient descent;
- fancy footwork with learning rate required to make this work;
- Viterbi decoding is used to compute the best output sequence.

Experimental Setup

- Train and test on CoNLL-2005 dataset (essentially a dependency parsed version of PropBank);
- word embeddings either randomly initialized or pretrained;
- pretrained embeddings used Bengio's Neural Language Model on English Wikipedia (995M words);
- vocabulary size 4.9M; embedding dimensionality 32;
- compare to feed-forward convolutional network;
- try different input features, different numbers of LSTM layers, and different hidden layer sizes.

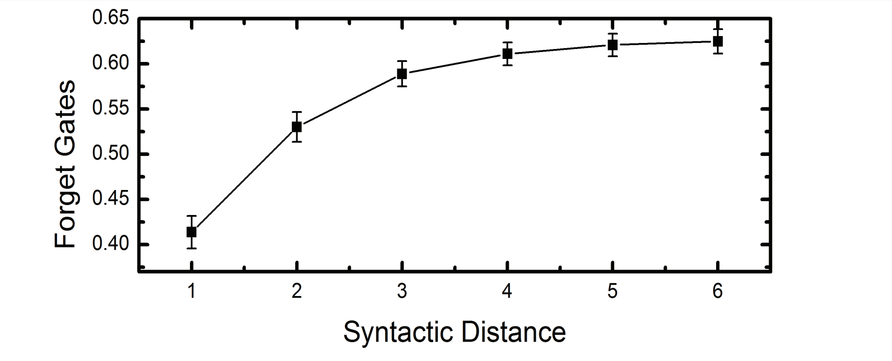
Results for CoNLL-2005 Dataset

Embedding	d	ctx-p	m_r	h	F1(dev)	F1
Random	1	1	n	32	47.88	49.44
Random	1	5	n	32	54.63	56.85
Random	1	5	y	32	57.13	58.71
Wikipedia	1	5	y	32	64.48	65.11
Wikipedia	2	5	y	32	72.72	72.56
Wikipedia	4	5	y	32	75.08	75.74
Wikipedia	6	5	y	32	76.94	78.02
Wikipedia	8	5	y	32	77.50	78.28
Wikipedia	8	5	y	64	77.69	79.46
Wikipedia	8	5	y	128	79.10	80.28
Wikipedia	8	5	y	128	79.55	81.07

d: number of LSTM layers; ctx-p: context length; m_r: region mark used or not; h: hidden layer size. Last row with fine tuning.

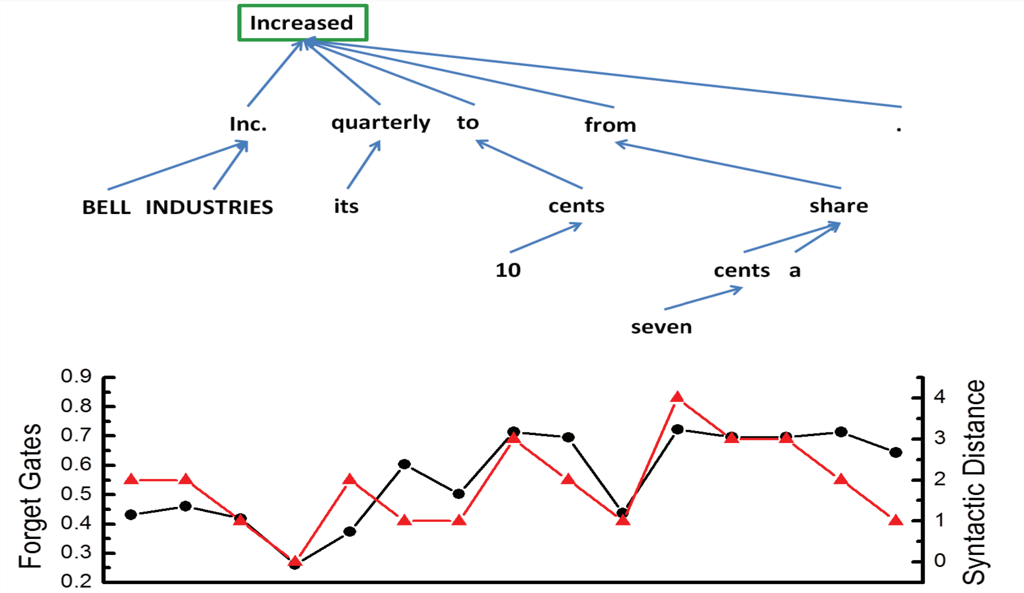
What the Model Learns (Maybe)

Model learns "syntax": it associates argument and predicate words using the forget gate:



Syntactic distance is the number of edges between argument and predicate in the dependency tree.

What the Model Learns (Maybe)



Summary

- Semantic role labeling means identifying the arguments (frame elements) that participate in a prototypical situation (frame) and labeling them with their roles;
- this provides a shallow semantic analysis that can benefit various NLP applications;
- SRL transitionally consists of parsing, frame element matching, feature extraction, classification;
- but it can also regarded as a sequence labeling task;
- Zhou and Xu use a deep bi-directional LSTM trained on embeddings to do SRL;
- no parsing needed, no handcrafted features;
- model may learn correlates of syntax anyway.